

**Vysoká škola báňská – Technická univerzita Ostrava**

**Fakulta bezpečnostního inženýrství**

**Katedra bezpečnostních služeb**

**Monitorování přítomnosti osob v objektu**

**Monitoring the presence of people in the building**

**Student:** Bc. Tomáš Olejník

**Vedoucí diplomové práce:** Ing. Petr Bitala, Ph.D.

**Studijní program:** Požární ochrana a průmyslová bezpečnost

**Studijní obor:** Technická bezpečnost osob a majetku

**Termín odevzdání diplomové práce:** 16. 4. 2021

## **Poděkování:**

Tímto bych chtěl poděkovat svému vedoucímu práce panu Ing. Petru Bitalovi, Ph.D., za odborné vedení, cenné rady a připomínky při zpracování mé diplomové práce.

## **Anotace**

Bc. Olejník, Tomáš, Monitorování přítomnosti osob v objektu. Diplomová práce. Ostrava, 2021. 66 stran, 1 příloha. Vysoká škola báňská – Technická univerzita Ostrava, Fakulta bezpečnostního inženýrství. Vedoucí diplomové práce Ing. Petr Bitala, Ph.D.

Tato diplomová práce se zabývá návrhem a realizací systému, sloužícího k monitorování přítomnosti osob v objektu za pomoci videoanalytických funkcí. Práce je rozdělena na dvě hlavní části. První část je věnována definici základních pojmů, stručné teorii zpracování obrazu a videoanalýzy. Druhá část je zaměřena na volbu vhodného technického řešení systému, jeho realizaci a následnému testování.

**Klíčová slova:** Videoanalýza, videoanalytické funkce, monitorování osob, open source platforma

## **Summary**

Bc. Olejník, Tomáš, Monitoring the presence of people in the building. Thesis. Ostrava, 2021. 66 pages, 1 attachment. Technical University of Ostrava, Faculty of safety engineering. Thesis supervisor Ing. Petr Bitala, Ph.D.

This thesis deals with design and implementation of a system used to monitor the presence of people in the building using video analysis functions. This thesis is divided into two main parts. The first part is devoted to definition of basic concepts, a brief theory of image processing and video analysis. The second part focuses on the choice of a suitable technical solution, implementation, and subsequent testing.

**Key words:** Video analysis, video analysis functions, monitoring of people, open source platform

## Obsah

Úvod.....	1
1 Rešerše.....	2
2 Základní pojmy.....	3
3 Zpracování obrazu .....	4
3.1 Počítačové vidění .....	4
3.2 Úrovně zpracování obrazu .....	4
4 Digitalizace.....	6
4.1 Systém zpracování digitálního obrazu .....	6
4.1.1 Nasnímání obrazu.....	7
4.1.2 Uložení .....	7
4.1.3 Zpracování.....	7
4.1.4 Komunikace.....	8
4.1.5 Zobrazení.....	8
5 Segmentace obrazu .....	9
5.1 Rozdělení metod segmentace obrazu .....	9
5.2 Metoda detekce hran .....	10
5.2.1 Cannyho hranový operátor .....	10
5.3 Metoda spojování hran .....	11
5.3.1 Lokální zpracování .....	11
5.3.2 Oblastní zpracování .....	11
5.3.3 Globální zpracování za použití Houghovy transformace .....	12
5.4 Metoda prahování.....	12
5.4.1 Globální prahování .....	13
5.4.2 Lokální prahování.....	14
5.5 Metoda šíření oblastí .....	14

5.6	Metoda dělení a spojování oblastí.....	14
6	Klasifikace objektů.....	16
6.1	Klasifikace objektů na základě tvaru .....	16
6.2	Klasifikace objektů na základě pohybu.....	17
6.3	Klasifikace objektů s využitím Neuronových sítí .....	17
6.3.1	Co je to neuronová síť .....	17
6.3.2	Princip funkce.....	18
6.3.3	Konvoluční neuronové sítě.....	19
7	Videoanalýza .....	22
7.1	Centralizovaná a decentralizovaná videoanalýza.....	22
7.2	Využití videoanalýzy v bezpečnostních aplikacích .....	23
7.2.1	Videoanalytické funkce .....	23
7.3	Využití videoanalýzy v ostatních oborech .....	26
8	Identifikace požadavků na technický systém a princip funkce .....	27
9	Volba technického řešení .....	28
9.1	Hardware .....	28
9.1.1	Počítač .....	28
9.1.2	Kamera .....	31
9.1.3	Napájecí adaptér .....	32
9.1.4	Paměťové médium .....	33
9.1.5	Kryt.....	34
9.2	Software .....	35
9.2.1	Operační systém .....	36
9.2.2	Programovací jazyk.....	37
9.2.3	Neuronová síť a programové knihovny.....	37
10	Realizace .....	39

10.1	Vytvoření krytu .....	39
10.2	Montáž.....	42
10.3	Instalace softwaru a spuštění zařízení .....	45
10.4	Programování .....	46
10.4.1	Implementace knihoven a vytvoření proměnných .....	47
10.4.2	Nastavení rozlišení a programování hranice .....	47
10.4.3	Implementace neuronové sítě, spuštění kamery a detekce.....	49
10.4.4	Zjištění třídy objektu a souřadnic .....	49
10.4.5	Počítání osob .....	50
10.4.6	Datový výstup.....	52
11	Testování.....	54
11.1	Testování videoanalytické aplikace.....	54
11.2	Testování tepelného namáhání .....	57
	Závěr .....	59
	Seznam zdrojů.....	60
	Seznam obrázků .....	63
	Seznam tabulek .....	65
	Seznam příloh .....	66

## **Seznam zkratek**

<b>FLOPS</b>	Počet operací v pohyblivé řádce za vteřinu
<b>FPS</b>	Snímková frekvence za vteřinu
<b>OS</b>	Operační systém
<b>Px</b>	Pixel
<b>SBC</b>	Jednodeskový počítač
<b>VSS</b>	Dohledové videosystémy

# Úvod

Ochranu objektu je možné realizovat různými bezpečnostními opatřeními, jako jsou například technická či režimová opatření. Jedním z možných technických opatření je využití kamerových systémů. Počátek používání kamerových systému v oblasti bezpečnostních aplikací se datuje do druhé poloviny 20. století. Ve svých počátcích byly využívány nejprve ve vojenských aplikacích, následně v průmyslu, v oblasti kontroly kvality a v oblastech nebezpečných lidskému zdraví. Poté následuje první použití v oblasti ochrany objektů, jedná se o banky či jiné peněžní ústavy, letiště apod.

Od svého prvního nasazení prošly dohledové videosystémy (dále jen VSS) dlouhým vývojem. Oproti svým počátkům se dnes s nimi setkáváme téměř na každém kroku. Jako reprezentativní příklady lze uvést například instalace v městských zástavbách, kde hlídají důležité dopravní uzly, parkoviště, ulice anebo náměstí. Jejich původní využití v objektech s nutností velkého stupně zabezpečení, se v současnosti rozšířilo na téměř jakýkoliv typ objektu. Příkladem mohou být průmyslové podniky, rozlehlé areály, obchodní centra, maloobchody, kanceláře, restaurace či rodinné domy. Současné VSS disponují kamerami kompaktních rozměrů, s vysokým rozlišením obrazu umožňující jejich využití téměř za jakýchkoliv podmínek. Díky dnešním technologiím mohou být kamery vybaveny termovizí, videoanalýzou či infračerveným přísvitkem, který umožňuje jejich provozování i za velmi špatných okolních světelných podmínek.

Vybavenost systémů VSS videoanalytickými funkcemi je dnes téměř samozřejmostí. Tyto funkce značně snižují nároky na obsluhu systému. Snižují spotřebu paměti na záznamových zařízeních a usnadňují obsluhu práci se záznamy. Videoanalytické funkce také do jisté míry automatizují systémy VSS. Jedná se o softwarové aplikace, které jsou schopné samostatně, automaticky vyhlásit poplachy, pracovat se záznamy, ukládat a zpracovávat metadata, či odesílat upozornění.

Cílem předložené diplomové práce je pomocí produktů open source platformy navrhnout a realizovat systém s analytickými funkcemi, který by sloužil k monitorování osob v objektu. První část práce se zabývá teorií, základy a metodami zpracování obrazu, které jsou nezbytné pro úspěšnou videoanalýzu. Druhá část práce je pak věnována samotnému návrhu a realizaci vlastního systému.



# 1 Rešerše

V této kapitole jsou uvedeny nejdůležitější zdroje použité pro vypracování této diplomové práce.

**HLAVÁČ, Václav a Miloš SEDLÁČEK. *Zpracování signálů a obrazů*. Praha: Vydavatelství ČVUT, 2000. ISBN 80-01-02114-9.**

Jedná se o českou publikaci, která se z počátku zaměřuje na základy zpracování jednorozměrných signálů. Ve své další části popisuje základy zpracování signálů dvojrozměrných neboli obrazových. Dále se věnuje tématu digitálních obrazů a pojmu digitalizace. V poslední řadě také tématům pořízení obrazu, filtrace šumu, poruch a detekce hran.

**KOLEKAR, Maheshkumar. *Intelligent Video Surveillance Systems: An Algorithmic Approach*. Boca Raton, Florida: Taylor & Francis Group, 2018. ISBN 978-1-4987-6711-8.**

Tato publikace se věnuje tématům inteligentních dohledových videosystémů. Kapitoly v úvodní části publikace jsou zaměřeny na základy zpracování obrazu a videa. Součástí těchto kapitol je popis různých metod zpracování obrazu, spolu s detailnějším popisem metody segmentace obrazu. Další kapitoly úvodní části jsou věnované problematice video komprese a analýze pohybu. Ve druhé části publikace jsou popsána témata klasifikace objektů v obraze, rozpoznání lidských aktivit a sledování objektů ve videosekvenci. Závěrečná část publikace je zaměřena na samotné VSS, jejich strukturu a využití v praxi.

**Convolutional Neural Networks. IBM [online]. 20.10.2020 [cit. 2021-02-13]. Dostupné z: [ibm.com/cloud/learn/convolutional-neural-networks](https://ibm.com/cloud/learn/convolutional-neural-networks)**

Webová stránka společnosti IBM, která definuje, co je to konvoluční neuronová síť. V textu je popsána struktura konvoluční neuronové sítě a popis její funkce. Dále jsou zde uvedeny různé druhy konvolučních neuronových sítí a v poslední řadě také příklady jejich využití v praxi.

## 2 Základní pojmy

V textu diplomové práce budou následující pojmy diskutovány v tomto významu.

**Dohledový videosystém** – Dohledový videosystém je systémem tvořeným kamerami, záznamovými, zobrazovacími, přenosovými a ovládacími zařízeními. Využitelný v oblasti bezpečnostních aplikací. [27]

**IP Kamera** – IP kamera je tvořena kombinací kamery s počítačem v jednom zařízení. Pro přenos obrazu využívá počítačovou síť. [24]

**Open source** – Jedná se o pojem, který slouží k označení počítačového softwaru. Tento software má volně přístupný zdrojový kód. Kdokoliv jej může studovat, libovolně využít a upravit pro další vývoj. [16]

**Operační systém** – Jedná se o řídicí program, který slouží k zajištění komunikace mezi uživatelem počítače a hardwarem. [11]

**Python** – Python je univerzální programovací jazyk, který je vyvíjen jako open source platforma. [21]

**Strojové učení** – Strojové učení je obor zaměřený na vytváření aplikací, které se učí z dat. Aplikace v průběhu času zlepšují svou přesnost, aniž by k tomu byly naprogramovány. [15]

**Hluboké učení** – Hluboké učení je podmnožinou strojového učení, které se zabývá neuronovými sítěmi. [6]

**Pixel** – Je nejmenším prvkem digitálního obrazu. Digitální obraz je tvořen z matice tisíců až miliónů pixelů. [3]

**Metadata** – Metadata jsou data o datech. Pomáhají hledat, uspořádat a porozumět datům. Jsou popisem a kontextem dat. [14]

### **3 Zpracování obrazu**

Zpracování obrazu lze chápat jako zpracování vícerozměrného signálu. Pod tímto termínem bude ve smyslu textu uvažován vícerozměrný signál jako obraz snímáný kamerou. Takovéto obrazy lze matematicky modelovat, a to prostřednictvím spojitě skalární funkce dvou nebo tří proměnných. Tato funkce se nazývá funkcí obrazovou. U statického dvojrozměrného obrazu se používá obrazová funkce dvou proměnných. U obrazů měnících se v čase, nebo obrazů trojrozměrných využijeme obrazovou funkci tří proměnných. Jednotlivé hodnoty obrazové funkce se rovnají určitým fyzikálním veličinám, jako je jas, barva apod. V dnešní moderní technice pracujeme s obrazy digitálními. U takových obrazů je obrazová funkce reprezentována maticí. Prvky matice pak tvoří jednotlivé obrazové prvky, nazývané se pixely. Obširnější pohled na téma zpracování obrazu nabízí obor počítačové vidění. [8]

#### **3.1 Počítačové vidění**

Počítačové vidění je obor, který se snaží o napodobení vidění člověka za pomoci technických prostředků. U počítačového vidění, je stejně jako u člověka potřeba ke zpracování a vyhodnocení obrazové informace inteligence, zkušeností a znalostí. Cílem je napodobit chování člověka a jeho rozhodování na základě informací vyplývajících z obrazu.

Zpracování dat obrazu je nejdůležitější částí počítačového vidění. Pomocí zpracování obrazu dochází k vytvoření výsledného modelu ze vstupních obrazových dat. Výsledný model pak slouží k porozumění a vyhodnocení obrazu počítačem. Na porozumění obrazu technickým zařízením mají vliv různé úrovně zpracování obrazu. [8]

#### **3.2 Úrovně zpracování obrazu**

Cílem správného porozumění obrazu počítačem je nalézt vztah mezi vstupním obrazem a vytvořeným výsledným modelem. Během procesu zpracování obrazu dochází k přechodu od vstupního obrazu k výslednému modelu a současně s tímto postupem dochází k nárůstu obrazových informací. Proces mezi vstupním obrazem a výsledným modelem lze rozdělit na několik úrovní zpracování obrazové informace.

Obecně lze rozdělit zpracování obrazu do tří úrovní (nízká, střední a vysoká). Definice těchto úrovní ovšem není přesně daná a hranice mezi jednotlivými úrovněmi se mohou lišit podle různých typů aplikací. [8], [9]

Do nízké úrovně zpracování obrazu lze zařadit digitální obrazy. Tyto obrazy mají pořád podobu nasnímaných obrazových dat, tedy jedná se o matice tvořené pixely s hodnotami jejich jasu. Obrazy zpracované na nejnižší úrovni jsou také výsledkem procesů předzpracování (jako je filtrace šumu či korekce) a slouží pro připravení obrazu k dalšímu zpracování. [8]

Střední úroveň zpracování obrazu tvoří tzv. příznaky a objekty. Příznaky jsou spojené části obrazu, které s určitou pravděpodobností tvoří jednotlivé objekty. Mohou to být dvojrozměrné oblasti či hranové segmenty, které se rovnají stěnám objektů. Objekty v obraze jsou výsledkem procesu segmentace. U náročných úloh s výskytem složitějších objektů může jít také o částečnou segmentaci, kdy interpretaci obrazu jako objektu je možno provést až po dalším zpracování.

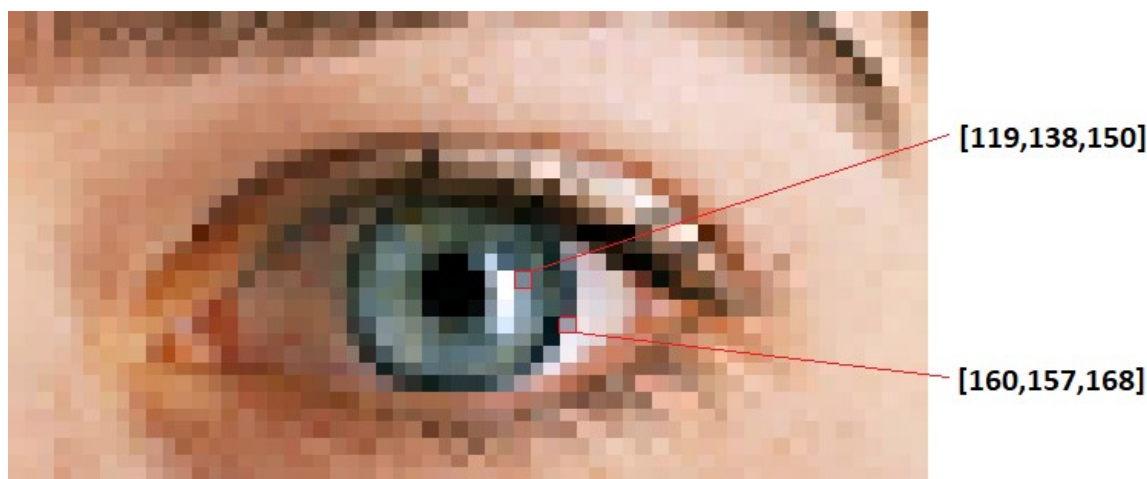
Vysokou úroveň zpracování obrazu reprezentují relační modely. Tyto modely vystihují kvalitativní i kvantitativní vlastnosti objektů nalezených v obrazech. Cílem vyšší úrovně zpracování je porozumění obsahu obrazů a videí. Využívá se zde metod a procesů rozpoznávání, umělé inteligence a videoanalýz. [8], [9]

## 4 Digitalizace

Díky nástupu digitálních technologií je možné pořídit digitální obraz pomocí různých zařízení jako jsou fotoaparáty, kamery, rentgenové zařízení, elektronové mikroskopy nebo radary. Zpracování digitálního obrazu se pak využívá v různých oborech jako je medicína, průmysl, obchod, bezpečnost či ve vojenských aplikacích. Cílem pro každý z těchto oborů je získání potřebných informací z digitálního obrazu.

Digitální obraz je vlastně maticí, jejíž řádkové a sloupcové indexy identifikují souřadnice bodu v obraze. Konkrétní bod v obraze má pak hodnotu, která odpovídá stupni šedi v tomto daném bodě. U barevných obrazů má konkrétní bod tři hodnoty, odpovídající intenzitě červené, zelené a modré (RGB barevný model). Jednotlivé body tvořící digitální obraz se nazývají pixely. [13]

Strukturu digitálního obrazu tvořenou maticí pixelů lze vidět na Obr. 1.



*Obr. 1 – Obrázek oka, s viditelnou maticí pixelů [autor, upraveno z [7]]*

### 4.1 Systém zpracování digitálního obrazu

Zpracování digitálního obrazu je proces využívající počítačové algoritmy a operace k získání informací z digitálních obrazů. Systém zpracování digitálních obrazů lze rozdělit na jednotlivé kroky. Těmito kroky jsou, nasnímání obrazu, uložení, zpracování, komunikace a zobrazení. [13]

### **4.1.1 Nasnímání obrazu**

Nasnímání obrazu představuje první krok v systému digitálního zpracování obrazu. Princip snímání obrazu IP kamerou lze popsat v několika následujících krocích. Světlo odražené ze snímaného objektu je soustřeďováno pomocí objektivu a optického filtru do obrazového snímače kamery (u IP kamer se jedná o CMOS čip). Obrazový snímač se skládá z dvourozměrného pole buněk, kde každá buňka představuje pixel ve výsledném obraze. Schopností obrazového snímače je převádět dopadající světlo na elektrické napětí. Toto napětí je následně převedeno na digitální signál. Více dopadajícího světla znamená vyšší napětí v daném bodě a naopak.

Digitální obrazy mohou být také získané pomocí konverze z obrazů analogových. Dále existují speciální metody snímání, jako je například trojrozměrné snímání obrazu za pomoci dvou a více snímacích zařízení, či použití satelitů k modelování různorodých povrchů. [13], [25]

### **4.1.2 Uložení**

Ukládání obrazových informací je důležité pro jejich dlouhodobé skladování a následnou využitelnost. Digitální obrazy a videosekvence mohou být zkomprimovány pro zmenšení jejich paměťových nároků a pro zvýšení přenosové rychlosti. Komprimaci lze rozdělit na bezztrátovou a ztrátovou.

Bezztrátová komprimace ušetří mnohem méně prostoru na paměťovém médiu bez výrazné změny dat digitálního obrazu. Oproti tomu ztrátová komprimace ušetří větší prostor na paměťovém médiu, ale za cenu ztráty obrazových dat. [13]

### **4.1.3 Zpracování**

Digitální obraz je zpracováván za pomoci různých metod, jako jsou matematické operace, metody zpracování signálů, nebo za pomoci metod počítačových. Výsledkem je zpracovaný digitální obraz, nebo parametry související s daným obrazem. U využití počítačových metod, systém zpracuje digitální obraz za pomoci počítačového algoritmu a vygeneruje výstupní obraz.

Mezi dnes běžně používané metody zpracování digitálního obrazu patří metoda rekonstrukce, mozaiková metoda, metoda segmentace, metoda rozpoznání obrazu a metoda vodoznaku.

Metoda rekonstrukce obrazu je technika, která vytváří dvojrozměrné a trojrozměrné obrazy z jednorozměrných projekcí. Běžně se využívá v tomografii nebo u magnetické rezonance. Metody rekonstrukce obrazu lze rozdělit na analytické a iterativní.

Mozaiková metoda vytváří digitální obraz zkombinováním obrazových informací z většího množství jiných obrazů. Příkladem může být mapování oblastí letícím dronem.

Metoda vodoznaku je technikou vkládání obrazových informací do digitálního obrazu za účelem ochrany vlastnictví. Metody vkládání vodoznaku lze rozdělit na viditelné a neviditelné.

Segmentace je proces, který vykreslí digitální obraz do smysluplných částí, které se následně využijí pro analýzu obrazu. Metoda převážně slouží k oddělení objektů nebo oblastí v obraze. Jelikož je metoda segmentace základem videoanalytických funkcí v bezpečnostních aplikacích, je podrobně rozebrána v samostatné kapitole.

Stejně jako segmentace také metody rozpoznání obrazu (klasifikace) nachází své využití v bezpečnostních aplikacích. Cílem těchto metod je rozpoznání konkrétních objektů v obraze, jako je člověk, tvář, automobil, zvíře apod. Metody rozpoznání obrazu jsou diskutovány podrobně v samostatné kapitole. [13]

#### **4.1.4 Komunikace**

Při přenosu digitálních obrazů je potřeba brát v úvahu přenosovou rychlost. Je-li k dispozici kanál s malou přenosovou rychlostí, je potřeba opět přistoupit ke komprimaci digitálního obrazu. Po přenosu obrazu dojde k jeho dekomprimaci a obnovení originálního obrazu. Digitální obrazy většinou obsahují spoustu nadbytečných informací, toho lze využít k dosažení komprimace a zrychlení přenosu. [13]

#### **4.1.5 Zobrazení**

Zobrazení je poslední částí systému zvaného zpracování digitálních obrazů. Pro realizaci zobrazení se využívají běžná zobrazovací zařízení, jako jsou počítačové monitory nebo tiskárny. [13]

## 5 Segmentace obrazu

Segmentace obrazu je technika, která rozděluje daný digitální obraz na smysluplné části s ohledem na konkrétní aplikaci. Tato technika slouží pro vyhledávání objektů a oblastí v obraze. Převážně se využívá pro analýzu obrazů a videoanalýzy. Jinými slovy je to proces, který přiřazuje označení každému pixelu v digitálním obraze na základě určitých vizuálních charakteristik. Soubor různých oblastí vytvořených segmentací obrazu společně tvoří celý obraz. Každý pixel v segmentované oblasti je podobný ostatním pixelům vyskytujícím se v téže oblasti, s ohledem na některé vlastnosti, jako je barva, intenzita, struktura a hloubka.

Segmentace obrazu je nedílným krokem pro pochopení jakéhokoliv digitálního obrazu. Zároveň s detekcí objektu dojde ke zjištění jeho vlastností, jako je velikost a tvar. U video aplikací se využívá segmentace k detekci pohybujících se objektů na scéně. [13]

### 5.1 Rozdělení metod segmentace obrazu

Metody segmentace obrazu lze obecně rozdělit na metody založené na podobnostech a metody založené na nesouvislostech.

Podobnostně založené metody fungují na principu rozdělení obrazu do oblastí, které jsou si podobné dle předem nastavených kritérií. Do této skupiny spadá metoda prahování, metoda šíření oblasti a metoda dělení a spojování oblastí. Za oblast v obraze považujeme skupinu spojených pixelů s podobnými vlastnostmi. Metody založené na prahování lze dále rozdělit na globální a lokální.

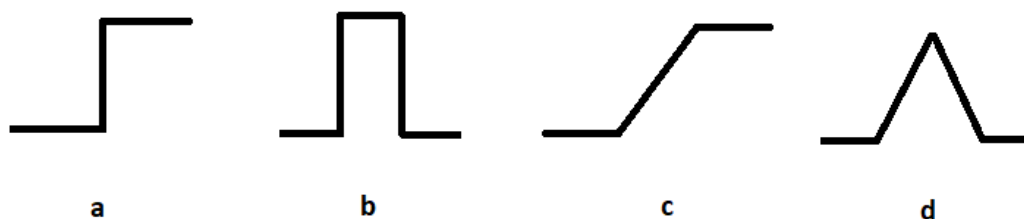
Metody založené na nesouvislostech fungují na principu rozdělení obrazu do odlišných oblastí podle skokových změn intenzity pixelů. V této skupině metod se nacházejí metoda detekce hran a metoda spojování hran. U těchto metod dochází k detekci hran, které znázorňují hranice oblastí, či objektů v obraze. Metoda spojování hran navíc spojuje přerušované hrany dohromady a tím dochází k lepším výsledkům segmentace. Tato metoda se dále dělí podle zpracování na lokální, oblastní a globální. Nejběžněji používanou metodou spojování hran, je metoda globálního zpracování založená na principu Houghovy transformace. [13]



## 5.2 Metoda detekce hran

Metoda detekce hran je proces vyhledávání nesouvislostí v obraze. Za nesouvislosti se považují náhlé změny v intenzitách pixelů, které charakterizují hranice objektu či oblasti. Pixely, jejichž intenzita obrazové funkce se vzhledem k okolí náhle změní, jsou nazývány hranové pixely. Tyto pixely jsou detekovány za pomoci výpočtu rozdílu intenzity v lokálních oblastech obrazu. K detekci těchto hranových pixelů dochází za pomoci algoritmů, které se nazývají hranové operátory.

Hrany v obraze lze rozdělit na, skokové, liniové, šikmé a střešní. Skoková hrana reprezentuje perfektní přechod z jedné oblasti do druhé. V tomto případě dochází ke skokové změně intenzity pixelů v obraze. Pokud je oblast v obraze velmi úzká, je jisté že má dvě hrany ve velmi těsné blízkosti. Takovému uspořádání se říká liniová hrana. Hrana šikmá nastává, pokud je přechod mezi dvěma oblastmi pozvolný. Pokud se nachází dvě šikmé hrany blízko vedle sebe podobně jako u liniového uspořádání hran, dochází k vytvoření střešní hrany. Jednotlivé typy hran lze vidět na Obr. 2. [13]



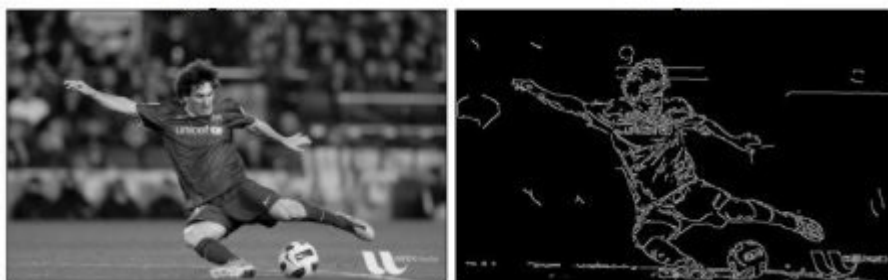
Obr. 2 – a) skoková, b) liniová, c) šikmá, d) střešní [13]

### 5.2.1 Cannyho hranový operátor

Cannyho hranový operátor je jedním z nejpoužívanějších hranových operátorů. Jedná se vlastně o vícestupňový algoritmus, který je schopen detekovat širokou škálu hran v obraze. Princip jeho funkce se skládá z následujících čtyř kroků.

Prvním krokem je redukce šumu v obraze. Toho je dosaženo za pomoci Gaussova filtru. Druhým krokem je provedení výpočtu gradientu intenzity. Tento krok slouží k zjištění intenzity hran a jejich směru. Třetím krokem je tzv. ztenčení. Jedná se o proces odstranění nežádoucích pixelů v okolí hrany, které nejsou její součástí. Posledním krokem je ověření pomocí prahových hodnot. Tato fáze rozhoduje o tom, které z nalezených hran jsou skutečně hranami a které nikoliv. Všechny hrany s gradientem intenzity větším, než je stanovená

prahová hodnota jsou potvrzenými hranami. Obraz segmentovaný metodou detekce hran za pomoci Cannyho hranového operátoru lze vidět na Obr. 3. [20]



*Obr. 3 – Původní obraz a obraz po segmentaci [20]*

### **5.3 Metoda spojování hran**

U metody detekce hran občas dochází ke stavu, kdy je detekovaná hrana nesouvislá či přerušovaná. Pro tyto případy je třeba použít techniku, která bude schopna přerušené hrany spojit. V praxi dochází k tvorbě přerušovaných hran z důvodu nerovnoměrného osvětlení snímané scény. Metoda spojování hran funguje na principu lokálního, oblastního a globálního zpracování. [13]

#### **5.3.1 Lokální zpracování**

Nejjednodušší způsob, jak lze spojit přerušené hrany je využít a spojit sousedící pixely s podobnými vlastnostmi. Poté co obraz projde metodou detekce hran, dochází k analýze lokálních oblastí (velikosti  $3 \times 3$  nebo  $5 \times 5$  pixelů). Pixely nalezené v lokálních oblastech, vyznačující se podobnými vlastnostmi jsou spojeny. Podobnost pixelů je stanovena na základě dvou klíčových vlastností. První vlastností je síla odezvy hranového operátoru použitého k detekci hranového pixelu. Druhou pak směr hranového operátoru. [13]

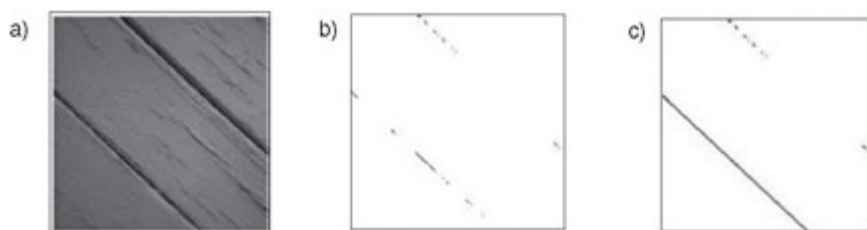
#### **5.3.2 Oblastní zpracování**

Tato technika se využívá k propojení pixelů v rámci větších oblastí v obraze, přičemž požadovaným výsledkem je aproximace hranice dané oblasti. Oblastního zpracování lze dosáhnout dvěma způsoby, a to aproximací funkční a polygonální. [13]

### 5.3.3 Globální zpracování za použití Houghovy transformace

Využití lokálního a oblastního zpracování je možné jen za podmínky, že máme alespoň částečné znalosti o tom, které pixely patří, ke kterým objektům. Pokud nemáme žádné informace o hranicích jednotlivých objektů či oblastí, využijeme zpracování globální. Toto zpracování funguje na principu Houghovy transformace.

Houghova transformace je technika sloužící k vyhledávání jednoduchých útvarů v obraze, jako jsou kružnice, elipsy a přímky. Principem funkce je hledání podmnožiny bodů v obraze, která co nejvíce odpovídá části jednoduchého útvaru například přímky. Každý bod na přímce je definován dvěma souřadnicemi. Realizace Houghovy transformace se provádí diskretizací obrazu. Každý prvek prostoru je pak tvořen dvojicí souřadnic. Na Obr. 4, lze vidět obraz zpracovaný metodou detekce hran a následně upravený metodou spojování hran na principu Houghovy transformace. [2]



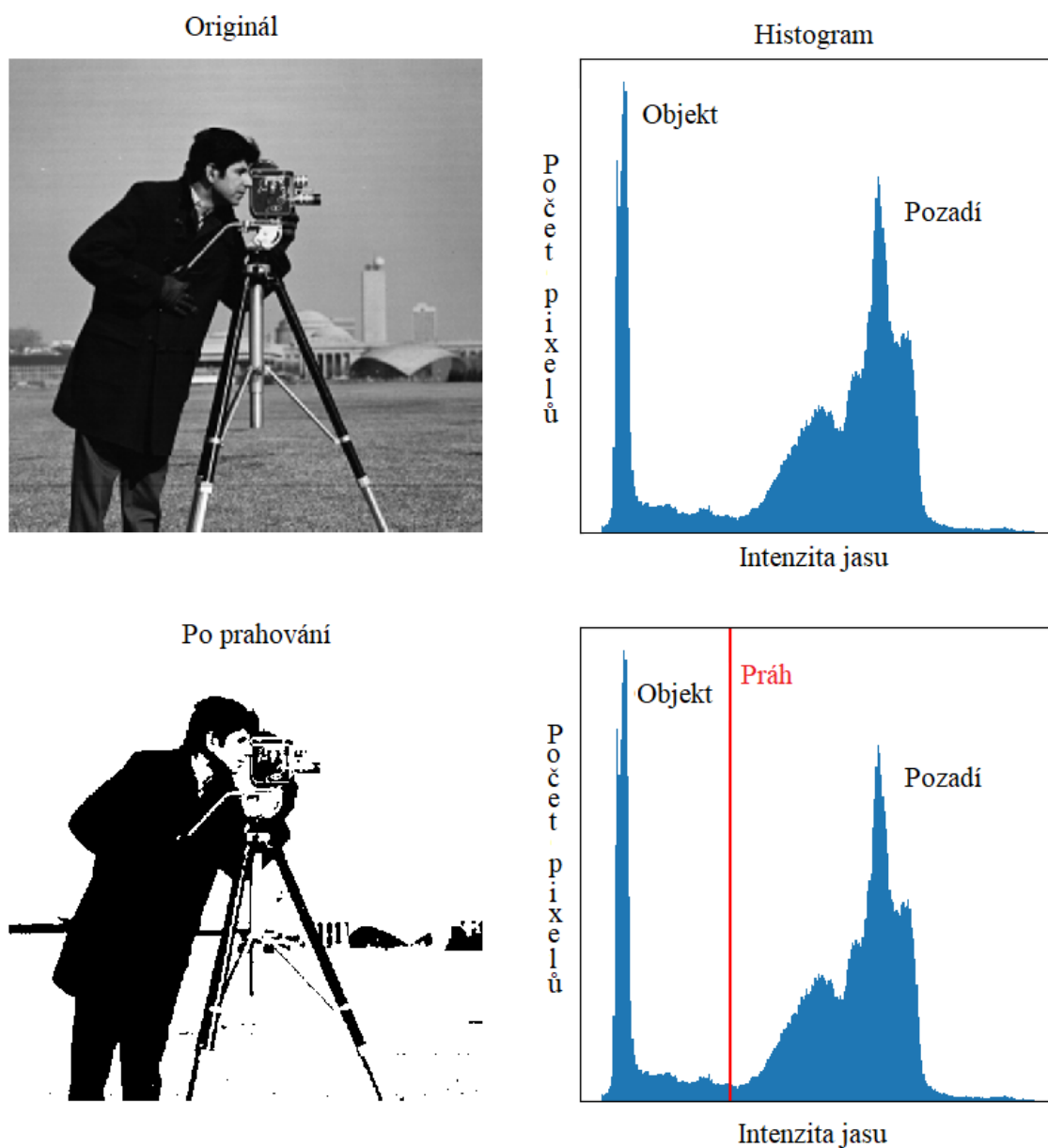
Obr. 4 – a) původní obraz, b) detekce hran, c) spojování hran [2]

### 5.4 Metoda prahování

Metoda prahování je nejjednodušší metodou segmentace obrazu. U této metody je vstupní obraz rozdělen do oblastí a objektů na základě hodnot intenzity jasu pixelů.

Metoda založená na prahování funguje na principu analýzy histogramu, který obsahuje informace o počtech pixelů v obraze s jejich hodnotami intenzit šedi. Analýzou histogramu dostaneme práh, přesněji tedy množinu prahů, za pomoci kterých lze obraz rozdělit na jednotlivé oblasti a objekty. Výsledkem prahování je binární obraz, který obsahuje jen dva různé typy pixelů, a to ty, které tvoří objekt a které jsou pozadím. Dle principu funkce lze rozdělit prahování na globální a lokální. [26]

Ukázka histogramu obrazu nasnímaného kamerou s detekovanou prahovou hodnotou, je k vidění na Obr. 5.



Obr. 5 – Histogram obrazu a jeho prahování [autor, upraveno z [13]]

#### 5.4.1 Globální prahování

Globální prahování využívá stejnou prahovou hodnotu pro všechny pixely v obraze. Tato metoda prahování funguje jen v případě, pokud histogram obrazu obsahuje dvě oddělené vrcholové hodnoty. Jedna odpovídající objektu a druhá pozadí v obraze. Pro kvalitnější výsledky segmentace je lepší využít lokálního prahování. [13]

### 5.4.2 Lokální prahování

Lokální prahování je založeno na výpočtu prahové hodnoty pro každý pixel zvlášť. Pokud je hodnota intenzity pixelu pod prahovou hodnotou, je přiřazen k pozadí. V opačném případě je přiřazen k objektu v popředí. [13]

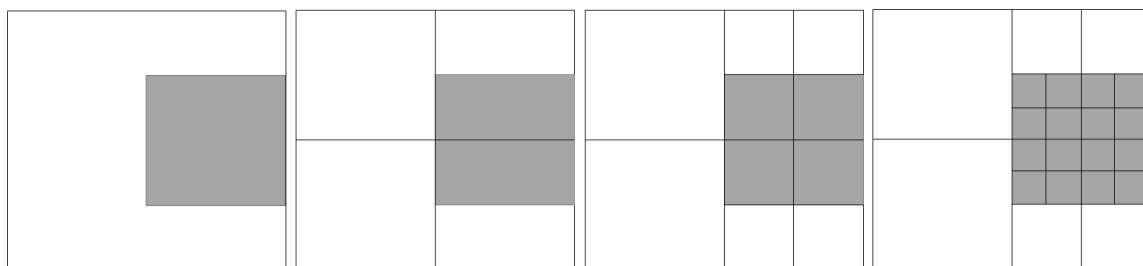
## 5.5 Metoda šíření oblastí

U některých obrazů s nekvalitním rozlišením je segmentace pomocí detekce hran dost obtížná a někdy i neproveditelná. V takovýchto případech se nabízí metoda šíření oblastí. Jak již bylo zmíněno, oblastí v obraze je skupina pixelů s podobnými vlastnostmi. Metoda šíření oblastí je principiálně velmi jednoduchou metodou. Její funkce je založena na shlukování pixelů nebo menších podoblastí do větších oblastí na základě předem definovaných kritérií.

Zpočátku je vybrána jedna sada počátečních bodů (pixelů). Na základě vybraného kritéria se otestují všechny sousední body. Pokud jsou sousední body podobné počátečním bodům, pak dochází k jejich spojení do oblasti. Tento proces je opakován, dokud nedojde k zastavení šíření. Kritériem podobnosti může být například intenzita, barva apod. [13]

## 5.6 Metoda dělení a spojování oblastí

Tato metoda je založena na stejnorodosti obrazu. Princip metody je následovný. Nejprve je vstupní obraz rozdělen na čtyři části, tyto části se otestují a zjistí se jejich stejnorodost. Části, které vyjdou jako nestejnorodé jsou opět rozděleny na čtyři menší části a znova otestovány. Toto dělení pokračuje do doby, dokud nejsou všechny části v obraze stejnorodé. Pokud mají stejnorodé části podobné vlastnosti jsou opět spojeny do jedné oblasti (objektu). Princip metody dělení a spojování oblastí je zobrazen na Obr. 6.



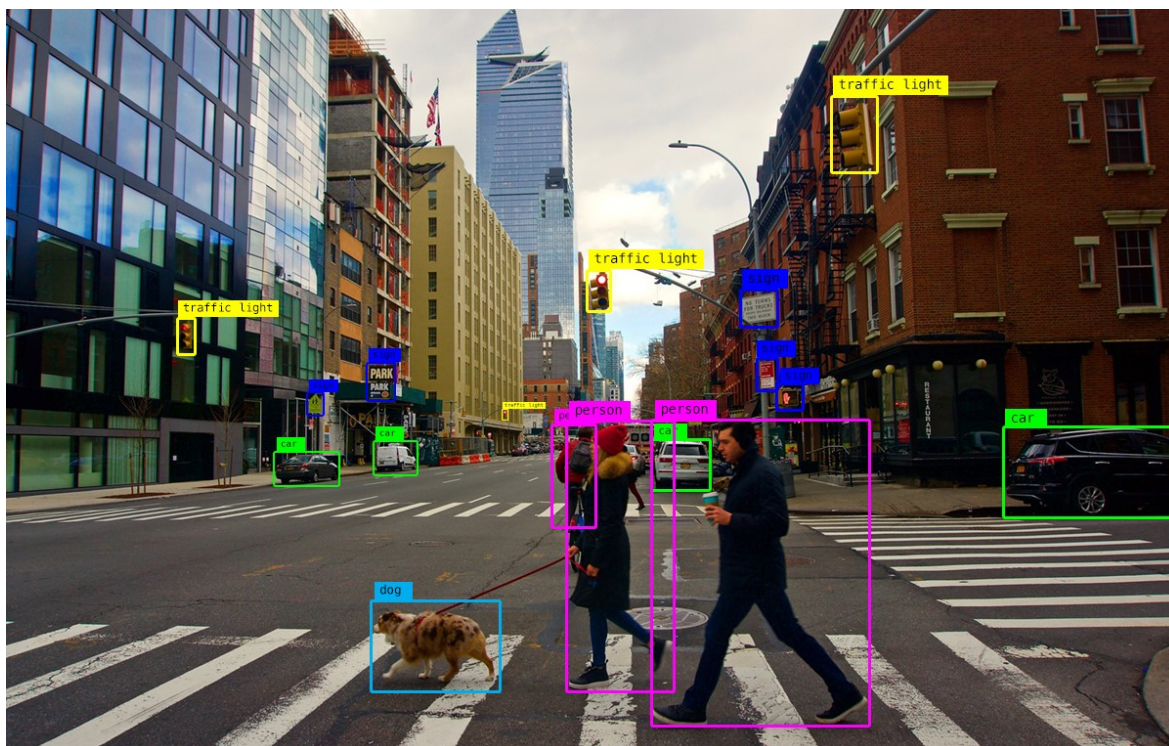
Obr. 6 – Princip metody dělení a spojování oblastí [autor, upraveno z [13]]

Výhodou je, že obraz může být dělen postupně podle požadovaného rozlišení. Počet úrovní dělení totiž volí sám uživatel. Kritéria stejnorodosti a spojování jsou také voleny uživatelem podle potřebné aplikace.

Nevýhodou dělení do čtyř částí je tvoření hranatých segmentů v obraze. Tvorbu hranatých segmentů lze snížit pomocí většího počtu dělení, ovšem pak roste výpočetní náročnost. [13]

## 6 Klasifikace objektů

Klasifikace nebo také rozpoznání objektů je proces, který je schopen detekované objekty v obraze či videosekvenci roztrždit do jednotlivých kategorií, jako jsou lidé, vozidla, zvířata apod. Ačkoliv je proces roztržení objektů pro člověka triviální, pro počítačový algoritmus je to úkol velmi náročný. Tento proces je nepostradatelnou součástí videoanalýzy. Na Obr. 7, lze vidět klasifikaci objektů v obraze probíhající v reálném čase.



Obr. 7 – Ukázka klasifikace objektů [19]

Klasifikace objektů je prováděna různými způsoby, běžně na základě tvaru a na základě pohybu. Klasifikace založená na tvarech využívá míru disperze a poměr stran objektu či oblasti jako klíčové vlastnosti pro jejich roztržení. Klasifikace založená na pohybu využívá pravidelnosti a opakovatelnosti lidských pohybů a aktivit. V posledním desetiletí se klasifikace objektů stále častěji provádí za pomoci neuronových sítí. [13]

### 6.1 Klasifikace objektů na základě tvaru

Ke klasifikaci objektů na základě tvaru se využívá disperze světla daného objektu. Z důvodu změny osvětlení, měřítka a pozice kamery ve scéně je zjištění disperze objektu náročným procesem.

Jelikož je tvar lidského těla oproti například vozidlu mnohem členitější a komplikovanější, objekty lidí na scéně budou vykazovat větší světelnou disperzi než vozidla. Pomocí světelné disperze lze tedy využít tvar objektu k jeho klasifikaci, například na člověka a vozidla ve scéně. Pro správné rozpoznání objektů u tvarových klasifikací je klíčový výběr správných vlastností objektu. [13]

## **6.2 Klasifikace objektů na základě pohybu**

Klasifikace na základě pohybu funguje v několika následujících krocích. Nejprve je oddělena silueta detekovaného objektu od pozadí scény. Následně je spočtena euklidovská vzdálenost od těžiště ke hraničnímu pixelu objektu. Za účelem klasifikace objektu jsou zjištěné vlastnosti porovnány se šablonami uloženými v databázi. Tyto šablony odpovídají různým lidským činnostem. Pohyby lidského těla mají většinou periodickou povahu, příkladem může být chůze a běh. Právě díky této vlastnosti je možné klasifikovat lidi od jiných objektů ve videu.

Klasifikace na základě pohybu se často využívá v bezpečnostních aplikacích, konkrétně u systémů VSS. V bezpečnostních aplikacích je většinou detekovaným objektem člověk nebo vozidlo, popřípadě zvíře. Díky klasifikaci objektu do třídy, je provádění následujících úkolů jako osobní identifikace, sledování objektu či rozpoznání činnosti mnohem efektivnější a přesnější. Z klasifikace objektů vychází techniky rozpoznání obličejů, které jsou také velmi užitečné v bezpečnostních aplikacích. [13]

## **6.3 Klasifikace objektů s využitím Neuronových sítí**

Nedávný vývoj v oblasti umělé inteligence a hlubokého učení, připravil cestu pro plnění složitých úkolů za pomoci využití neuronových sítí. V poslední době se neuronové sítě využívají k řešení různých problematik a dosahují velmi dobrých výsledků v širokém zástupu oborů. Zejména v oblasti klasifikace, segmentace obrazu a celkově v oboru počítačového vidění. V těchto oblastech dochází k detekci, lokalizaci a klasifikaci objektů v obrazech a videosekvencích, za využití různých variant neuronových sítí. [13]

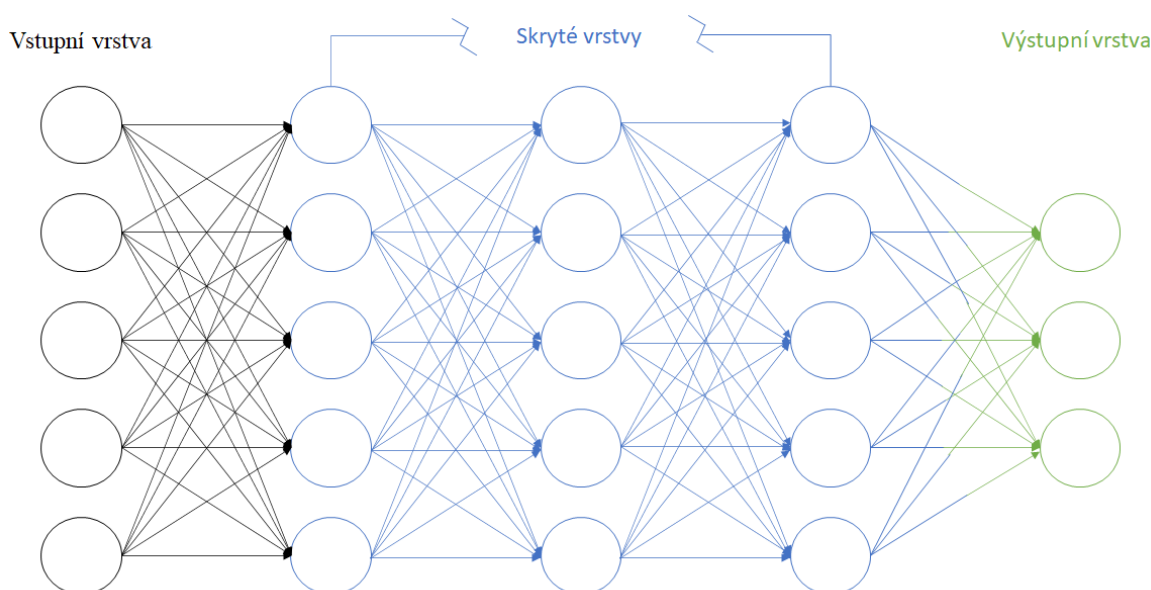
### **6.3.1 Co je to neuronová síť**

Neuronové sítě, také známé jako umělé neuronové sítě nebo simulované neuronové sítě, jsou podmnožinou strojového učení a jsou srdcem algoritmů hlubokého učení. Jejich



název a struktura je inspirována funkcí lidského mozku. Neuronové sítě napodobují způsob, jakým si biologické neurony předávají vzájemně signály.

Umělé neuronové sítě jsou složeny z vrstev uzlů obsahujících vstupní vrstvu, jednu nebo více skrytých vrstev a výstupní vrstvu. Každý uzel neboli umělý neuron je spojen s jiným neuronem a má přidruženou váhu a prahovou hodnotu. Pokud je výstup kteréhokoli uzlu nad zadanou prahovou hodnotou, je tento uzel aktivován a odesílá data do další vrstvy sítě. V opačném případě se žádná data do další vrstvy nepřevádí. Strukturu umělé neuronové sítě lze vidět na Obr. 8.



Obr. 8 – Struktura umělé neuronové sítě [autor, převzato z [17]]

Neuronové sítě využívají ke své funkci tréninková data, z těch se časem učí a zlepšují svou přesnost. Jakmile jsou tyto učící se algoritmy vyladěny, stávají se mocnými nástroji v oboru umělé inteligence, které nám umožňují klasifikovat, seskupovat a zpracovávat data vysokou rychlostí. Úkoly rozpoznávání řeči nebo rozpoznávání obrazů mohou ve srovnání s manuální identifikací lidskými odborníky trvat minuty oproti hodinám. Jednou z nejznámějších neuronových sítí je internetový vyhledávací algoritmus Google. [17]

### 6.3.2 Princip funkce

Problematika neuronových sítí je velmi obsáhlé a komplikované téma a pro potřeby této diplomové práce je popis principu jejich funkce zobrazen.

Prvním krokem je nastavení vstupní vrstvy neuronové sítě. Jakmile je nastavena, jsou ji přiřazeny váhy. Tyto váhy pomáhají stanovit důležitost jakékoli vstupní proměnné. Přičemž vstupy s větší váhovou hodnotou přispívají k výstupu významněji než jiné vstupy. Následně jsou všechny vstupy vynásobeny příslušnými váhami a sečteny. Výsledek je předán aktivační funkci, ta určuje výstup. Pokud hodnota výstupu překročí danou prahovou hodnotu, dojde k aktivaci uzlu a ten předá data do další vrstvy v síti. To má za následek, že výstup jednoho uzlu se stane vstupem dalšího uzlu. S vyšším počtem vrstev je neuronová síť schopna dělat složitější rozhodnutí v závislosti na předchozích rozhodnutí nebo vrstev.

Když začneme uvažovat o praktičtějších případech použití pro neuronové sítě, jako je rozpoznávání nebo klasifikace obrazů, využijeme k trénování algoritmu učení s učitelem. Během učení s učitelem vyhodnocujeme přesnost algoritmu pomocí funkce nákladů. Tato funkce se běžně označuje jako střední kvadratická chyba. Pomocí nákladové funkce dochází v každém tréninkovém procesu ke korekci nastavených prahových hodnot a vah. Cílem je co nejvíce minimalizovat nákladovou funkci, abychom dosáhli co největší přesnosti neuronové sítě. [17]

### **6.3.3 Konvoluční neuronové síť**

Neuronové sítě můžeme rozdělit do několika různých typů, které se používají pro různé účely. Z hlediska bezpečnostních aplikací jsou nejvyužívanějším typem konvoluční neuronové sítě, které se používají pro úkoly klasifikace, rozpoznávání obrazů a počítačové vidění. [17]

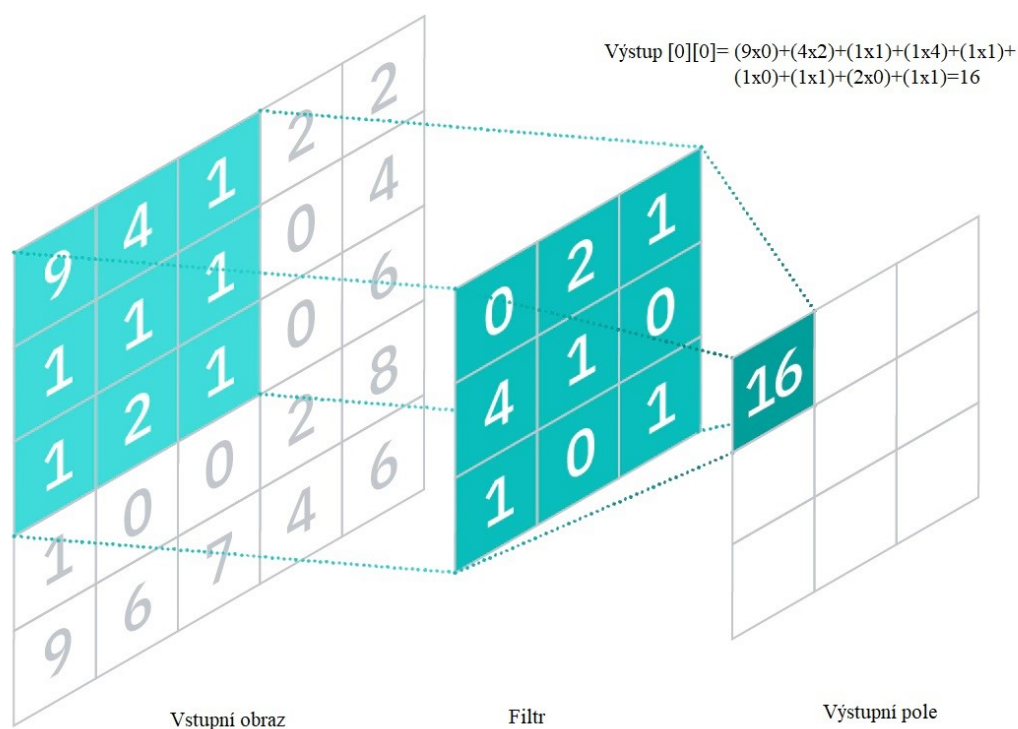
Konvoluční neuronové sítě se od ostatních neuronových sítí odlišují podle jejich vynikajícího výkonu při práci se vstupy ve formě obrazů, řeči a zvuku. Obsahují tři hlavní typy vrstev, kterými jsou:

- konvoluční vrstva;
- sdružovací vrstva;
- plně připojená vrstva.

První vrstvou konvolučních sítí je vrstva konvoluční. Zatímco po konvolučních vrstvách mohou následovat další konvoluční vrstvy nebo vrstvy sdružené, plně připojená vrstva je poslední vrstvou sítě. S každou vrstvou je konvoluční síť složitější a je schopna identifikovat a pracovat s většími částmi obrazu. Dřívější vrstvy se zaměřují na jednoduché

funkce, jako jsou barvy a hrany v obraze. Jak obrazová data procházejí vrstvami konvoluční sítě, dochází k rozpoznávání větších prvků nebo tvarů objektů, dokud nakonec nedojde k identifikaci zamýšleného objektu.

Konvoluční vrstva je základním stavebním kamenem konvolučních sítí, a právě v ní dochází k většině výpočtů. Vyžaduje několik komponentů, kterými jsou vstupní data, filtr a mapa funkcí. Vstupními daty je digitální barevný obraz, který je tvořen maticí pixelů. Filtr je dvojrozměrné pole vah, které představuje část obrazu. Filtry se mohou velikostí lišit, ale obvykle se jedná o matici velikosti 3×3 pixelů. Velikost filtru také určuje velikost zkoumaného pole. Při aplikaci filtru na oblast v obraze dochází mezi vstupními pixely a filtrem k výpočtu bodového součinu. Výsledek je poté převeden do výstupního pole. Následně se filtr posouvá krok za krokem a neustále opakuje svůj proces, dokud neprojde celým obrazem. Konečným výstupem ze série bodových součinů je mapa funkcí nebo také konvoluční funkce. Na Obr. 9, je k vidění funkce filtru. [4]



Obr. 9 – Použití filtru na vstupní obraz v konvoluční vrstvě sítě [4]

Jak lze vidět na Obr. 9, každá výstupní hodnota na mapě funkcí se nemusí spojit s každou hodnotou pixelu na vstupním obrázku. Postačí, když je výstupní hodnota spojena pouze ke konkrétnímu vstupnímu poli, kde byl filtr použit.

Hodnoty vah ve filtru zůstávají stejné po celou dobu jeho pohybu po obraze. Mohou se ovšem měnit během procesu učení. Celkem existují tři parametry filtru, které ovlivňují velikost výstupu a které je třeba nastavit před zahájením učícího procesu neuronové sítě. Jedná se o počet filtrů, krok a výplň.

Počet filtrů určuje, kolik filtrů bude použito a tím pádem udává počet výstupních mapových funkcí. Krok je vzdálenost či počet pixelů, kterými se filtr pohybuje po vstupní matici. Výplň je parametr, který se využívá, když velikost filtru neodpovídá vstupnímu obrazu.

Další vrstvou v konvolučních neuronových sítích je sdružovací vrstva. Ta provádí zmenšení rozměrů a snižuje počet parametrů na vstupu. Podobně jako u konvoluční vrstvy i vrstva sdružovací používá ke svým operacím filtr. Rozdíl je v tom, že tento filtr netvoří žádné váhy. Místo toho používá na hodnoty ve vstupním poli agregační funkci a pomocí té naplňuje výstupní pole. Existují dva hlavní typy sdružování, a to sdružování maximální a průměrné.

U maximálního sdružování vybírá filtr maximální hodnoty ze vstupních oblastí a posílá je do výstupního pole. Naopak u průměrného sdružování filtr vypočítává průměrnou hodnotu v rámci oblasti vstupního pole a tu posílá do výstupního pole.

Ve sdružovací vrstvě dochází ke ztrátám mnoha informací, avšak má to i řadu výhod. Dochází ke snížení složitosti, zvyšuje se efektivita sítě a omezí se riziko přehlcení daty.

V poslední, plně připojitelné vrstvě dochází ke spojení každé hodnoty pixelu vstupního obrazu k výstupním hodnotám konvolučních a sdružovacích vrstev. Tato vrstva provádí detekci či klasifikaci obrazu na základě funkcí extrahovaných předchozími vrstvami. Plně připojitelné vrstvy obvykle využívají ke klasifikaci vstupních hodnot tzv. aktivační funkci nazývanou softmax. Jedná se o normalizovanou exponenciální funkci, jejíž výsledkem je hodnota pravděpodobnosti od 0 do 1. [4]

## 7 Videoanalýza

Systémy VSS vyprodukují obrovské množství záznamů. Kvůli nedostatku času a zdrojů však většina těchto záznamů není nikdy reálně zhlédnuta nebo analyzována. V důsledku toho dochází k přehlédnutí bezpečnostních incidentů a pozdním reakcím na právě nastalé bezpečnostní incidenty. Tyto problémy hrály velkou roli ve vývoji videoanalýzy.

Videoanalýza je softwarový proces, který automaticky sleduje a zaznamenává informace o tom, co se děje ve videu. Ukládá a pracuje s metadaty, které lze použít k popisu osob, vozidel a dalších objektů detekovaných ve videosekvenci. Získané informace lze následně využít pro realizaci různě orientovaných akcí, např. vyhlášení poplachu, zahájení nahrávání apod. Nezbytné metody, na kterých je videoanalýza založena a se kterými pracuje, jako je segmentace a klasifikace, byly popsány v předešlých kapitolách.

Videoanalýza přináší do systémů VSS spoustu výhod. Především jde o úsporu nákladů. Díky využití videoanalýzy dochází k menšímu zatížení sítě a paměťového uložení videosystému. Šetří také čas, jelikož značně usnadňuje a zkracuje dobu vyhledávání a práci se záznamy. Díky automatickému sledování vykazují VSS vybavené videoanalýzou větší efektivitu a jsou schopné předcházet kriminální činnosti. [28]

### 7.1 Centralizovaná a decentralizovaná videoanalýza

Centralizovaná videoanalýza se využívala u starších analogových systémů. V těchto instalacích bylo video z kamer přeneseno do zařízení nazývaného digitální videorekordér (DVR), kde proběhla analýza videa. To bohužel znamenalo, že musel být přenesen celý záznam, často tvořen mnoha hodinami videa, které neobsahovalo z pohledu bezpečnosti žádný zajímavý obsah. Tento způsob videoanalýzy nadměrně zatěžoval síť a uložení. Navíc bylo potřeba výkonných a nákladných zařízení pro zpracování obrovského objemu dat.

V dnešních VSS se využívá decentralizovaná videoanalýza. U těchto systémů dochází k videoanalýze přímo v jednotlivých kamerách. Jedná se o mnohem efektivnější a flexibilnější architekturu. Odpadá nutnost pořizovat nákladná centralizovaná zařízení a nedochází k nadměrné zátěži uložení a sítě během přenosu. [28]

## 7.2 Využití videoanalýzy v bezpečnostních aplikacích

Dnešní nejrozšířenější oblastí, kde dochází k použití videoanalýzy, jsou bezpečnostní aplikace. V rámci této oblasti umožňují nástroje videoanalýzy realizaci širokého spektra videoanalytických funkcí. Ty mají za úkol především detekovat neobvyklé události bezpečnostního charakteru a následně na ně vhodným způsobem reagovat. Některé z těchto videoanalytických funkcí jsou uvedeny v následující podkapitole. [1], [29], [10]

### 7.2.1 Videoanalytické funkce

Mezi nejběžněji se vyskytující videoanalytické funkce využívané v oblasti bezpečnosti patří:

- detekce pohybu;
- překročení hranice;
- narušení oblasti;
- počítání lidí;
- detekce davu;
- detekce chybějících a opuštěných předmětů;
- rozpoznání poznávacích značek;
- rozpoznání tváře.

Uvedený výčet funkcí není konečný. V praxi se jich vyskytují desítky a spolu s rozvojem technických možností a parametrů komponentů systémů vznikají rovněž funkce nové. V následujících odstavcích jsou vybrané videoanalytické funkce podrobněji popsány.

Jednou z nejzákladnějších videoanalytických funkcí je detekce pohybu. Tato funkce umožní kameře detekovat pohyb ve videu a následně spustit nahrávání, upozornit obsluhu či spustit poplach. Detekce pohybu značně šetří nároky na paměťové uložení VSS oproti režimu nepřetržitému nahrávání. Zároveň se jedná o funkci, která slouží jako základ pro celou řadu jiných videoanalytických funkcí.

Detekce překročení hranice je analytickou funkcí využívanou například v perimetrické ochraně nebo na vlakových nástupištích apod. Kamera je schopna detekovat pohybující se objekt (člověk, vozidlo), který překročí definovanou hranici a vhodně reagovat. Na Obr. 10, lze vidět přelezení plotu osobou a tím překročení střežené hranice kamerou.



*Obr. 10 – Videoanalytická funkce překročení hranice [5]*

Detekce narušení oblasti poskytuje automatické sledování obvodu definované oblasti. Při proniknutí objektu do zabezpečené oblasti dojde k reakci systému vhodným způsobem, nejčastěji vyhlášením poplachu. Typickým příkladem oblastí hlídaných touto analytickou funkcí jsou prostory za pokladnou, prostory u exponátů v muzeích a galeriích. Příklad narušení oblasti lze vidět na Obr. 11.

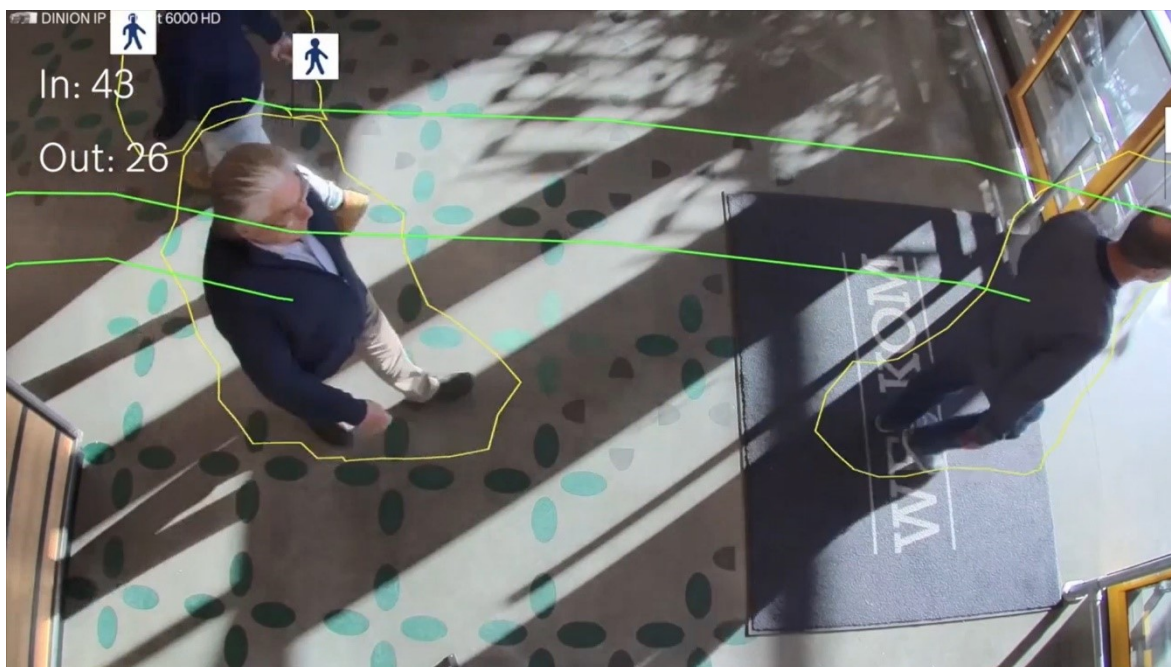


*Obr. 11 – Detekce narušení oblasti [5]*

Často používanou analytickou funkcí v oblasti bezpečnosti je počítání lidí. Tato funkce, jak napovídá její název, monitoruje počet osob, které prošli daným bodem ve scéně



viz Obr. 12. Běžně se využívá u vstupů do objektů, jako jsou školy, obchody, banky, letiště a věznice. Díky této funkci je k dispozici neustálý přehled o počtu osob, které se pohybují v objektu či zabezpečené zóně. Tato informace je velmi užitečná například při evakuaci osob z objektu.



*Obr. 12 – Videoanalytická funkce počítání lidí [5]*

Další analytická funkce vhodná pro bezpečnostní aplikace je detekce davu. Funkce umožňuje kameře vyhlásit poplach při překročení stanoveného procenta lidí ve scéně (obsazenost). Ideální využití nachází ve veřejných prostorech, jako jsou metra nebo nákupní střediska, s cílem zachování veřejné bezpečnosti.

Detekce chybějících předmětů je analytickou funkcí, díky které je kamera schopna detekovat chybějící objekt ve scéně. Podobnou analytickou funkcí je detekce opuštěných předmětů. U této funkce kamera rozpozná opuštěný předmět ve scéně, který zde někdo zanechal (například taška na letišti). Detekci opuštěných předmětů lze vidět na Obr. 13.





*Obr. 13 – Detekce opuštěných předmětů [5]*

Automatické rozpoznání poznávacích značek umožňuje kamery přechít a zaznamenat poznávací značku vozidla. Analytickou funkci lze využít k automatizaci přístupového systému vozidel, zaznamenávání a informování o příjezdu a odjezdu vozidel z objektu nebo ke kontrole na dálnicích či hraničních přejezdech.

Rozpoznání tváře je analytická funkce, která slouží k rozpoznání známých a neznámých jedinců na základě detekce a rozpoznání jejich tváře. Rozpoznání tváře lze využít k automatickému přístupovému systému. Dnes se tato analytická funkce využívá již běžně například na fotbalových a jiných sportovních stadiónech. [1], [29], [10]

### **7.3 Využití videoanalýzy v ostatních oborech**

Již zmíněnou oblastí využitelnosti videoanalýzy je oblast bezpečnosti. Kromě této oblasti ovšem nachází videoanalýza uplatnění v celé řadě oborů. Jedním z nich je například zdravotnictví, kde lze využít videoanalýzu k detekci poruch autistického spektra u dětí, případně k detekci pádu starších a nemohoucích pacientů. Dalším oborem, kde se videoanalýza osvědčila je doprava, kde analyzuje provoz, monitorující dopravní zácpy, detekuje vozidla jedoucí v protisměru, dopravní nehody nebo monitoruje parkovací plochy. Své využití nachází videoanalýza také v oblasti obchodu a marketingu, kde dochází ke sběru dat za účelem optimalizace prodeje a organizace obchodu. [1]

## **8 Identifikace požadavků na technický systém a princip funkce**

Před samotnou realizací vlastního technického systému pro monitorování osob v objektu bylo nejprve nutné si stanovit, jak bude výsledný systém vypadat, na jakém principu bude fungovat a jaké musí splňovat požadavky.

Vlastní technický systém je založen na principu videoanalytických funkcí. Jedná se tedy o zařízení složené z počítačového hardwaru, vybaveného softwarovou aplikací a kamery. Všechny zmiňované části jsou integrovány do jednotného celku a tvoří tak neoddělitelnou součást výsledného technického řešení zařízení. V zařízení dochází pomocí softwarové aplikace založené na umělé neuronové síti k videoanalýze kamerou snímané scény. Vzhledem k tomu, že videoanalýza je prováděna přímo v samotném zařízení, kde dochází také ke snímání obrazu, jedná se o videoanalýzu decentralizovanou.

Pro úspěšné monitorování osob v objektu systém splňuje následující požadavky.

Počítačový hardware je kompaktních rozměrů a má dostatečný výpočetní výkon k provádění videoanalytických funkcí. Zároveň je vybaven dostatečnou konektivitou k připojení ovládacích či zobrazovacích zařízení, ethernetové sítě a samotné kamery. V poslední řadě je schopen být napájen pomocí adaptéru ze zásuvky nebo z autonomního zdroje.

Kamera podobně jako počítačový hardware disponuje kompaktními rozměry. Je schopna pořizovat jak diskrétní snímky, tak kontinuální videosekvence. Je vybavena čipem schopným zpracovávat obraz s dostatečným rozlišením pro potřeby videoanalýzy a je kompatibilní s použitým počítačovým hardwarem.

Veškerý software použitý pro realizaci je ve formě open source platformy.

Splněním těchto výše uvedených požadavků je systém schopen za pomoci videoanalýzy úspěšně detekovat osobu a poznat, zdali vchází či vychází z objektu nebo do objektu (zóny v objektu). Zároveň je systém schopen uchovávat informaci o počtu osob vyskytujících se uvnitř.

## 9 Volba technického řešení

Obsah této kapitoly je věnován výběru technického vybavení použitého k realizaci vlastního systému pro monitorování osob v objektu. V následujících podkapitolách jsou uvedeny a popsány jednotlivé součásti vybrané k výsledné realizaci. Kapitola je rozdělena na část hardwarovou a softwarovou.

### 9.1 Hardware

Hardwarové řešení zahrnuje všechny fyzické komponenty využívané v rámci realizace technického řešení. V tomto případě jde o následující:

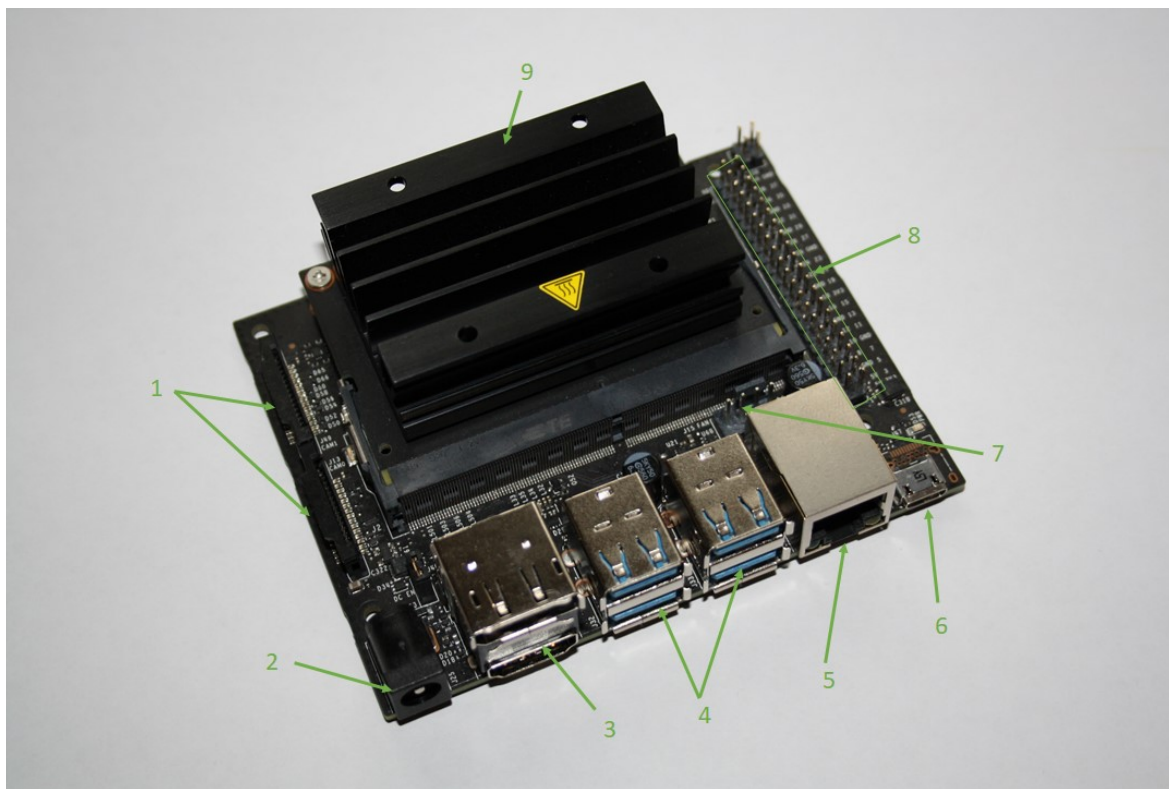
- počítač;
- kameru;
- paměťovou kartu;
- napájecí zdroj;
- kryt zařízení.

#### 9.1.1 Počítač

Vzhledem ke stanoveným požadavkům na kompaktní rozměry nešlo využít klasických stolních počítačů a notebooků. K řešení se proto nabízejí jednodeskové počítače (dále jen SBC). Tato zařízení se vyznačují velmi malými rozměry (velikosti kreditní karty). Problémem u většiny SBC je ovšem nedostatek výpočetního výkonu k provedení videoanalýzy. Proces videoanalýzy je totiž velmi náročný na grafický procesor počítače. Běžné SBC typu Raspberry Pi nebo Arduino nedisponují dostatkem grafického výkonu k provádění videoanalýzy v reálném čase. Proto bylo potřeba vybrat takový SBC, který se specializuje na tyto typy aplikací. S přihlédnutím na výše zmíněné požadavky, byl pro realizaci vybrán SBC nazývaný se Jetson Nano od společnosti NVIDIA.

Jetson Nano je na své malé rozměry vybaven velmi výkonným grafickým procesorem. Pomocí svého grafického procesoru je zařízení schopné se specializovat na aplikace neuronových sítí, detekce a klasifikace objektů v obrazu a jiné aplikace počítačového vidění a umělé inteligence. Díky tomu je ideálním komponentem pro technický systém zaměřený na monitorování lidí v objektu. Jetson Nano je zařízením typu SBC to znamená, že oproti běžným stolním počítačům má veškeré komponenty

umístěny na jedné desce plošných spojů. Počítač dosahuje rozměrů 100×80×29 mm. Vzhled počítače Jetson Nano je k vidění na Obr. 14, s číselným označením jednotlivých částí a komponent.



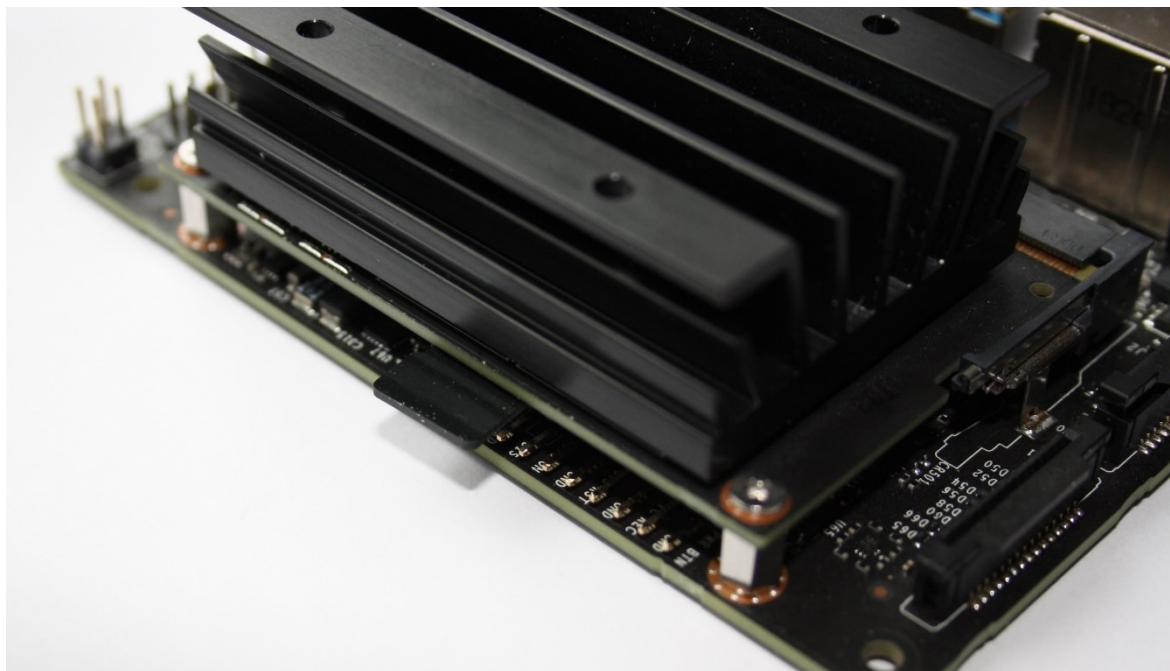
*Obr. 14 – Jednodeskový počítač NVIDIA Jetson Nano [autor]*

Popisy k označeným částem počítače jsou uvedeny v Tab. 1.

*Tab. 1 – Popisy jednotlivých částí počítače NVIDIA Jetson Nano [autor]*

Označení	Popis
1	2× CSI konektor pro připojení kamer
2	1× Souosý konektor pro připojení napájení 5 V
3	1× HDMI + 1× DisplayPort konektor pro připojení zobrazovacích zařízení
4	4× USB 3.0 port pro připojení ovládacích a paměťových zařízení
5	1× Ethernet port pro připojení do sítě
6	1× MicroUSB port pro připojení napájení 5 V nebo přenos dat
7	1× konektor pro připojení aktivního chlazení
8	40 pinová sběrnice pro připojení rozšiřujících zařízení
9	1× pasivní kovový chladič

Nevýhodou počítačů typu SBC je absence paměťového média. Jetson Nano není v tomto případě výjimkou. Důvodem chybějícího paměťového uložení jsou kompaktní rozměry tohoto typu počítačů. U Jetson Nano stejně jako u mnoha jiných SBC je tento problém řešen slotem pro paměťovou kartu. Ten se u Jetson Nano nachází na spodní části desky počítače. Na Obr. 15 lze vidět paměťovou kartu připravenou k zasunutí do slotu, který je skryt pod pasivním chladičem počítače.



*Obr. 15 – Paměťová karta připravená k zasunutí do slotu [autor]*

Detailněji jsou technické parametry počítače Jetson Nano prezentovány v Tab. 2. Výkonost počítače je uvedena v jednotkách FLOPS. [12]

*Tab. 2 – Technické parametry počítače Jetson Nano [12]*

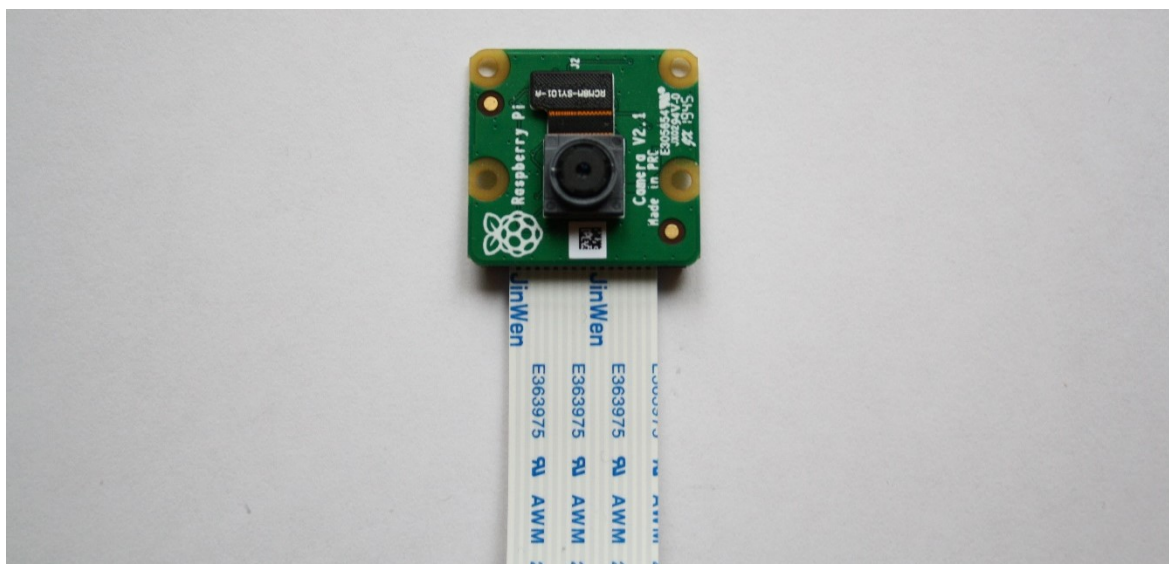
Parametr	Hodnota/Typ
<b>Výkon umělé inteligence (AI)</b>	472 GFLOPS
<b>Grafický procesor (GPU)</b>	128 jádrový NVIDIA Maxwell
<b>Centrální procesor (CPU)</b>	4 jádrový ARM Cortex – A57 Mpcore
<b>Paměť RAM</b>	4 GB 64-bit LPDDR4
<b>Sít'ování</b>	10/100/1000 BASE-T Ethernet
<b>Výkon</b>	5 W/10 W



### 9.1.2 Kamera

Primárním měřítkem pro výběr kamery byla vhodná konektivita se zvoleným počítačovým hardwarem. K počítači Jetson Nano lze připojit kameru pomocí konektorů CSI nebo USB. Neméně důležitým faktorem byly také co nejmenší možné rozměry kamery. Nejlépe tyto kritéria splňovala kamera Raspberry Pi V2. Jak již napovídá její název, jedná se o kameru stejnojmenného výrobce počítačů. Kamera je vybavena konektorem typu CSI a lze ji tedy připojit i k počítači Jetson Nano.

Objektiv kamery spolu s obrazovým snímačem je uložen na malé desce plošných spojů, kde se nachází veškerá elektronika. Z desky dále vychází CSI konektor o délce 15 cm, kterým lze kameru připojit k tomu uzpůsobeným zařízením. Celkově kamera dosahuje rozměrů 25 mm×20 mm×9 mm. Vzhled vybrané kamery lze vidět na Obr. 16.



*Obr. 16 – Kamera Raspberry Pi V2 [autor]*

Kamera je schopna natáčet videosekvence i snímat jednotlivé fotografie. Srdcem kamery je čip Sony IMX219, který má na starosti snímání obrazu. Pro uchycení kamery slouží čtyři montážní otvory umístěné v okrajové části desky plošných spojů. Napájení zajišťuje počítačový hardware, ke kterému je kamera připojena pomocí CSI konektoru.

Nevýhodou kamery je neschopnost snímání za špatných okolních světelných podmínek. Kamera totiž není vybavena infračerveným přísvitem ani jinou technologií umožňující plnou funkčnost za těchto okolností. [23]

Ostatní vlastnosti a technické parametry kamery jsou sepsány v Tab. 3.

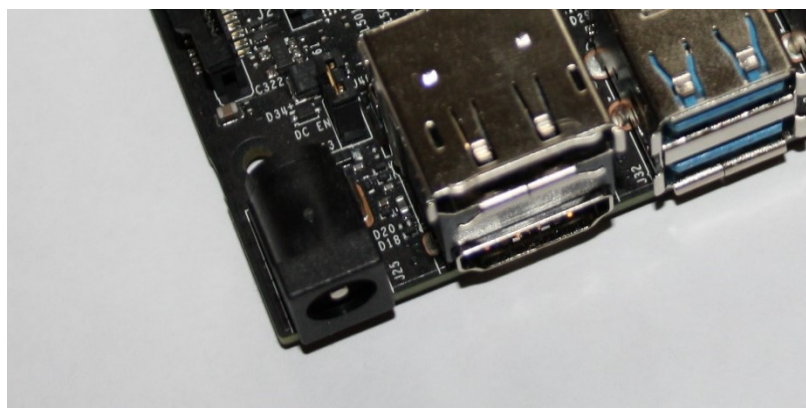
Tab. 3 – Technické parametry kamery Raspberry Pi V2 [23]

Parametr	Hodnota
Video FPS	1080p/30 FPS, 720p/60 FPS
Rozlišení fotografie	3280×2464 px
Čip	8 Mpx Sony IMX219 CCD

### 9.1.3 Napájecí adaptér

Napájecí adaptér slouží k napájení počítačového hardwaru elektrickou energií. Samotná kamera je napájena přímo z počítače pomocí CSI konektoru a nepotřebuje tedy pro napájení vlastní adaptér.

Počítač Jetson Nano funguje na napětí 5 V a lze jej napájet dvěma způsoby pomocí MicroUSB nebo souosým konektorem o rozměrech 5,5/2,1 mm. Z těchto dvou způsobů byl zvolen způsob napájení přes souosý konektor. Toto rozhodnutí bylo založeno převážně na dvou důvodech. Prvním je fakt, že MicroUSB konektor lze jinak využít i pro připojení různých externích zařízení a pro přenos dat. Druhým důvodem je možnost většího proudového odběru přes souosý konektor (až 4 A), oproti MicroUSB (max 2 A). Díky možnosti většího odběru proudu, vzniká možnost připojení různých externích zařízení, které nedisponují vlastním zdrojem energie. Příkladem může být připojení aktivního chlazení (ventilátor) nebo různých senzorů pro detekci fyzikálních veličin. Souosý konektor se na desce počítače nachází v levém spodním rohu viz Obr. 17.



Obr. 17 – Umístění souosého napájecího konektoru [autor]

Pro napájení počítače Jetson Nano byl vybrán napájecí adaptér Sunny model SYS1449-2005-W2E. Tento napájecí adaptér je vlastně usměrňovačem, který slouží pro připojení do zásuvky běžné elektrické sítě o střídavém napětí 230 V, 50 Hz. Střídavé vstupní napětí pak usměrní na výstupní stejnosměrné napětí o velikosti 5 V, kterým je napájeno koncové zařízení, v tomto případě počítač Jetson Nano. Maximální výstupní výkon adaptéru dosahuje 20 W. Vzhled vybraného napájecího adaptéru je k vidění na Obr. 18.



*Obr. 18 – Napájecí adaptér [autor]*

#### **9.1.4 Paměťové médium**

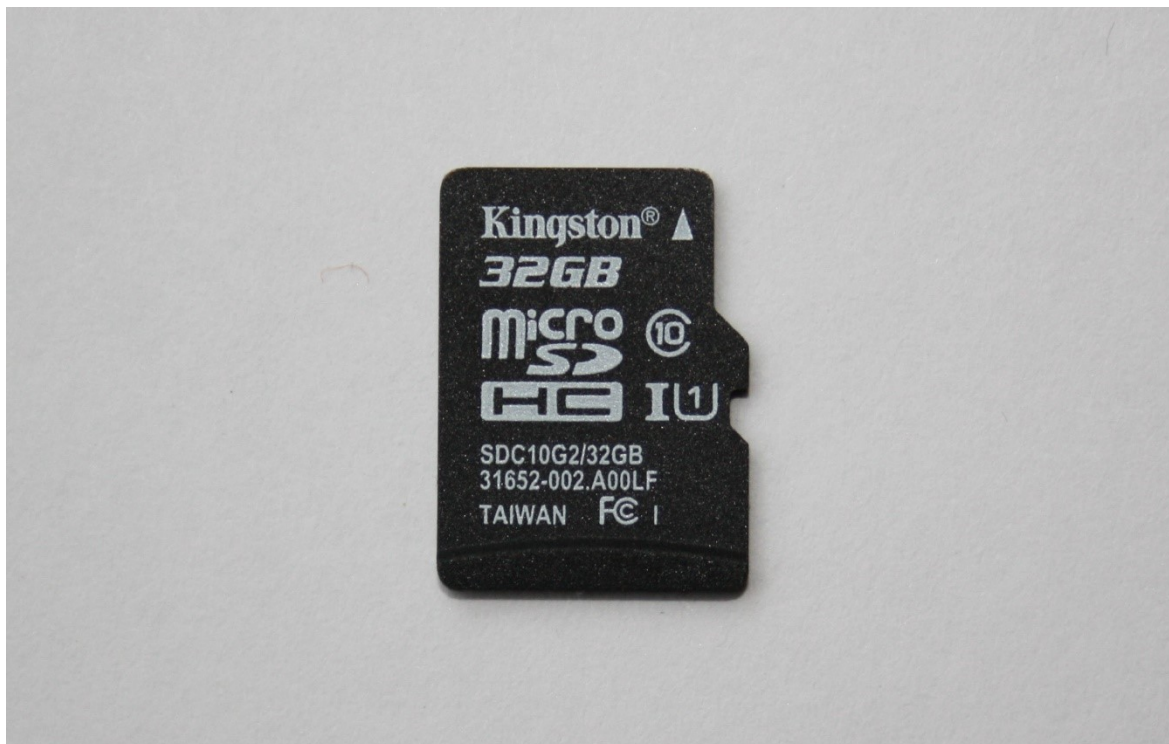
Pro realizaci vlastního systému je pořízení paměťového média nezbytně nutné. Na paměťovém médiu je totiž uložen veškerý software použitý pro realizaci, spolu se samotnou videoanalytickou aplikací a jejími nashromážděnými daty.

Počítač Jetson Nano umožňuje použít paměťové médium ve formátu paměťových karet. Konkrétně se jedná o typ paměťových karet MicroSDHC. Jde o běžně využívaný typ paměťové karty, která nachází své využití u různorodých zařízení (například digitální fotoaparáty). MicroSDHC karty disponují kompaktními rozměry a širokou škálou možných paměťových kapacit.

Po zjištění, kolik paměti zabere software potřebný pro realizaci zařízení, byla vybrána karta značky Kingston s kapacitou paměti 32 GB a přenosovou rychlostí 100 MB/s.



Tato velikost paměťového uložště je dostatečná pro uložení potřebného softwaru, realizaci vlastní videoanalytické aplikace a zároveň poskytuje dostatek zbylé paměti pro uložení nasbíraných dat. Vzhled vybrané paměťové karty lze vidět na Obr. 19.



*Obr. 19 – Paměťová karta MicroSDHC [autor]*

### 9.1.5 Kryt

Na trhu se vyskytuje celá řada krytů pro SBC vytvořených z různých materiálů. Žádný z nich ovšem nevyhovoval stanoveným požadavkům na použití počítače Jetson Nano s kamerou Raspberry Pi V2. Z tohoto důvodu bylo přistoupeno k řešení navrhnout a vyrobit kryt vlastní. Vlastní kryt byl zhotoven přesně na míru dle potřeb realizovaného zařízení a k jeho výrobě byla použita 3D tiskárna nacházející se na Fakultě bezpečnostního inženýrství.

Kryt byl zhotoven z plastového materiálu a skládá se ze dvou částí (hlavní části a vrchního krytu). Hlavní část je vybavena distančními sloupky pro uchycení počítače a vrchního krytu pomocí šroubových spojů. Dále je opatřena otvorem pro připojení konektorů a větracími otvory pro průchod vzduchu. Vrchní kryt obsahuje distanční sloupky pro uchycení kamery, montážní otvory pro uchycení krytu k hlavní části, otvor pro objektiv kamery a v poslední řadě také větrací otvory. Detailní popis způsobu vytvoření krytu a jeho montáž je popsána v kapitole Realizace.

Vzhled jednotlivých částí krytu lze vidět na Obr. 20 a Obr. 21.



*Obr. 20 – Hlavní část krytu [autor]*



*Obr. 21 – Vrchní část krytu [autor]*

## 9.2 Software

Do této části spadá veškeré programové vybavení počítače, které bylo využito pro realizaci vlastního systému. Jedná se tedy o:

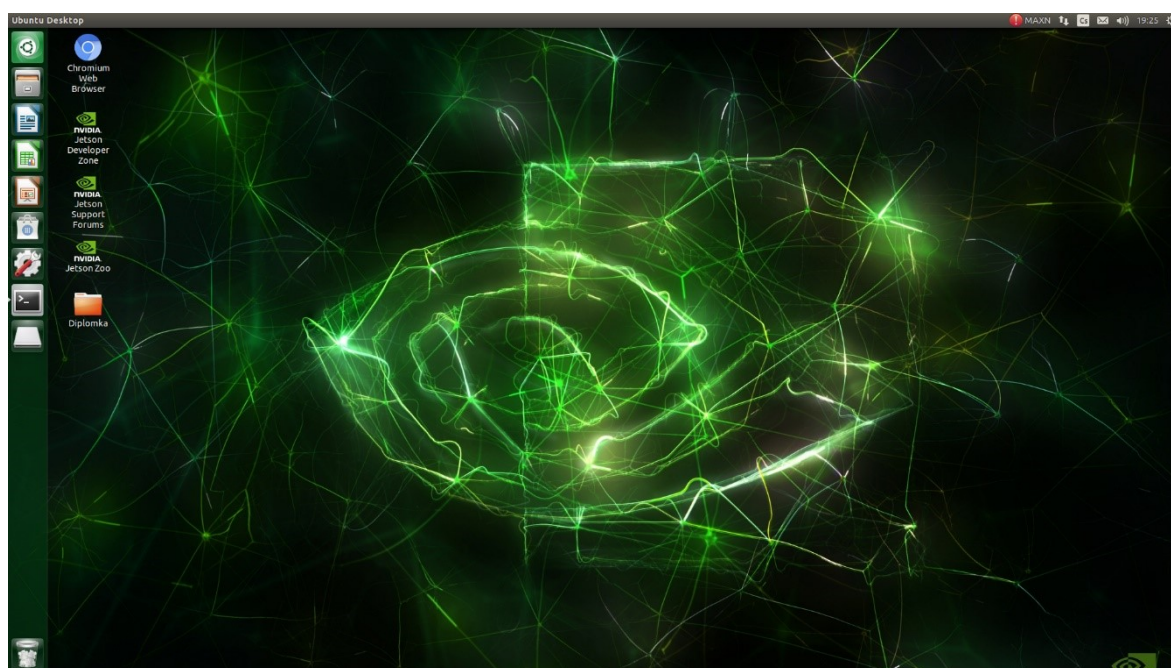
- operační systém;

- programovací jazyk;
- neuronovou síť;
- programové knihovny.

### 9.2.1 Operační systém

Operační systém (dále jen OS) je řídicím programem, pomocí kterého uživatel ovládá celý počítač. Jedná se tedy o nezbytný software, bez kterého nelze systém pro monitorování osob realizovat. Počítač Jetson Nano se s OS nedodává a je nutné jej pořídit zvlášť a následně nainstalovat na paměťové médium. Stanoveným požadavkem bylo, aby všechen využitý software byl ve formě open source platformy. V této formě se nabízí hned několik vhodných OS. Patří mezi ně například Ubuntu, Chrome, Android či Fedora.

Pro realizaci vlastního systému byl zvolen OS Ubuntu. Tento řídicí program má linuxovou architekturu a je volně stažitelný a šířitelný. Pro snadnější ovládání uživatelem je systém vybaven grafickým uživatelským rozhraním typu GNOME. Systém Ubuntu je z počátku vybaven spoustou užitečných programů, mezi něž patří textový a tabulkový editor, internetový prohlížeč, průzkumník souborů, příkazový řádek pro ovládání příkazy a spousta dalších. Drtivá většina programů, které lze spustit na systému Ubuntu, jsou rovněž poskytovány ve formě open source stejně tak jsou poskytovány i aktualizace samotného OS. Vzhled grafického rozhraní systému Ubuntu lze vidět na Obr. 22.



Obr. 22 – Grafické rozhraní operačního systému Ubuntu [autor]



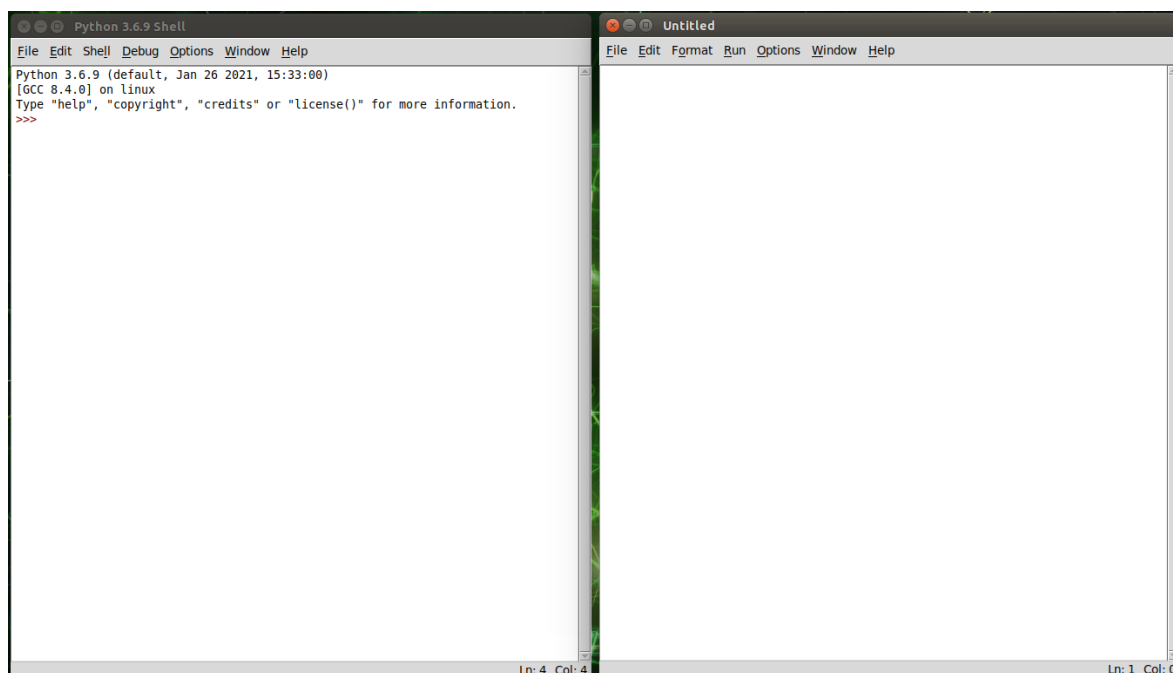
## 9.2.2 Programovací jazyk

Pro tvorbu vlastního systému bylo potřeba si zvolit programovací jazyk, ve kterém byla softwarová aplikace napsána. Jako nejznámější a nejvyužívanější open source programovací jazyky se nabízí Python, C, C++, C #, Java či PHP.

K realizaci byl vybrán programovací jazyk Python. Důvodem tohoto výběru byly již určité zkušenosti s tímto programovacím jazykem a také fakt, že velká část použitého OS je napsaná právě v tomto jazyce.

Python je interpretovaný objektově orientovaný programovací jazyk s dynamickou sémantikou. Jedná se o velmi oblíbený jazyk pro rychlý vývoj aplikací. Disponuje jednoduchou a snadno naučitelnou syntaxí. Je velmi čitelný, a proto snižuje náklady na údržbu programu. Python je k dispozici ve své zdrojové nebo binární formě zdarma pro všechny platformy a lze jej volně distribuovat. [22]

Vývojové prostředí, ve kterém je psána vlastní aplikace se nazývá IDLE Python. Vzhled tohoto prostředí jazyka Python lze vidět na Obr. 23.



Obr. 23 – Vývojové prostředí IDLE Python [autor]

## 9.2.3 Neuronová síť a programové knihovny

K realizaci systému pro monitorování osob v objektu byl použit model neuronové sítě, který je součástí sady vývojových nástrojů nazývajících se NVIDIA TensorRT.

Tato vývojová sada je poskytována ve formě open source platformy a je optimalizována na počítače Jetson Nano. Součástí vývojové sady je několik modelů neuronových sítí, které jsou vyvinuty se zaměřením na detekci a klasifikaci objektů v obraze.

K realizaci byl použit model neuronové sítě označující se SSD-MobileNet-v2. Tato neuronová síť je zaměřena na detekci objektů v obraze a jejich následnou klasifikaci až do 91 tříd.

Spolu s modely neuronových sítí jsou v rámci TensorRT k dispozici i programové knihovny. Programové knihovny obsahují předepsané funkce, které slouží při programování a tvorbě aplikací. Vývojová sada TensorRT nabízí pro práci s neuronovými sítěmi dvě knihovny, jetson-inference a jetson-utils. Obě knihovny nabízejí užitečné funkce využitě při realizaci systému.

## 10 Realizace

Následující text je věnován chronologickému popisu vytvoření vlastního systému sloužícího pro monitorování osob v objektu. V následujících podkapitolách je postupně uveden podrobný popis vytvoření krytu pro zařízení, montáž jednotlivých částí, instalace a spuštění nezbytného softwaru a samotné programování.

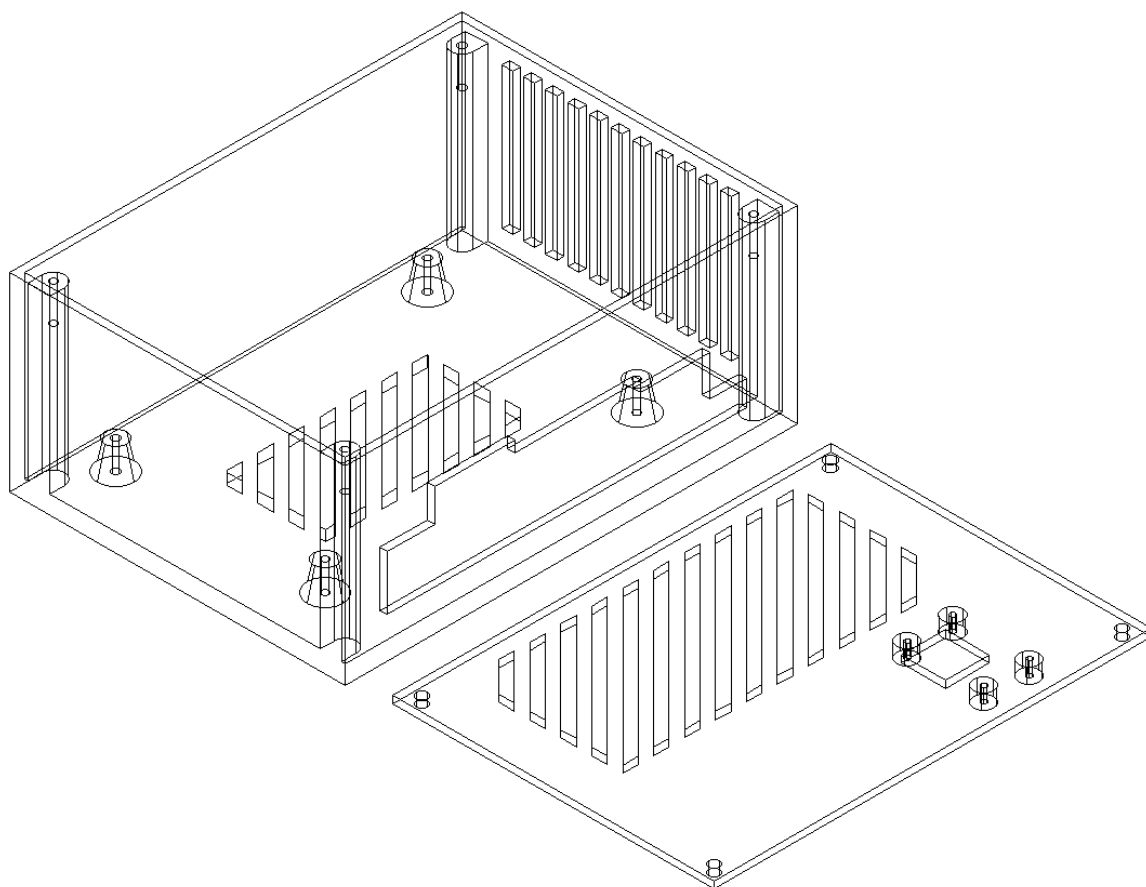
### 10.1 Vytvoření krytu

Prvním krokem při výrobě krytu bylo změření rozměrů vybraných hardwarových komponent za pomoci posuvného měřítka. Následně byl kryt skládající se ze dvou částí nakreslen pomocí softwarového programu. Nakreslený model byl následně vytištěn pomocí 3D tiskárny. Během procesu vytváření krytu byly použity programy PrusaSlicer (open source platforma) a AutoCAD (studentská verze).

Jako první byla nakreslena hlavní část krytu. Konkrétně tedy její spodní deska o šířce 3 mm, spolu se čtyřmi distančními sloupky pro uchycení počítače. Následně byly nakresleny stěny hlavní části krytu o šířce 2 mm. Spolu se stěnami byly vytvořeny další čtyři distanční sloupky, které slouží pro uchycení vrchní části. Tyto sloupky vycházejí ze spodní desky a končí 2 mm pod vrchní hranou stěn. V pravé boční stěně byl vytvořen podélný otvor sloužící pro připojení konektorů k počítači. V poslední řadě byly u hlavní části krytu vytvořeny otvory ve formě žebrování pro průchod vzduchu. Tyto otvory se nacházejí na spodní desce a zadní stěně krytu. Distanční sloupky jsou vybaveny kruhovými otvory o poloměru 1 mm a hloubky 8 mm. Otvory v distančních sloupcích slouží pro realizaci šroubových spojů.

Po zkrácení hlavní části krytu následovala část druhá. Jedná se vlastně o vrchní desku krytu o šířce 2 mm, která je vybavena kruhovými montážními otvory o poloměru 1,5 mm. Vrchní deska byla nakreslena tak, aby byla při montáži zapuštěna do hlavní části krytu. Po zkrácení montážních otvorů byl do desky vytvořen otvor o rozměrech 12×12 mm. Tento otvor je průzorem pro objektiv kamery. Následně byly kolem otvoru nakresleny čtyři distanční sloupky. Ty se nacházejí na spodní straně desky a slouží pro uchycení kamery.

Jako poslední byly nakresleny otvory pro větrání jdoucí téměř po celé délce vrchní desky. Výsledný model obou částí krytu nakreslený v programu AutoCAD lze vidět na Obr. 24.



*Obr. 24 – Model krytu nakreslený v programu AutoCAD [autor]*

Po nakreslení modelu krytu byl soubor uložen ve formátu STL. V tomto formátu byl soubor otevřen v programu PrusaSlicer. PrusaSlicer je programem pro 3D tiskárny. Slouží pro nastavení tisku a tiskového materiálu tzv. filamentu. Program lze také využít pro kreslení vlastních modelů.

Jako materiál byl vybrán plastový filament PLA. Jedná se o biologický plastový materiál, který se vyznačuje svou pevností a pružností. Veškeré potřebné nastavení tiskového materiálu a 3D tisku samotného je zobrazeno na Obr. 25.

**Výška vrstvy**

- Výška vrstvy:  mm
- Výška první vrstvy:  mm nebo %

**Svislé stěny**

- Perimetry:  (minimálně)
- Spirálová váza: ☐

Doporučená tloušťka stěny objektu pro výšku vrstvy 0.15 a 2 perimetry: 0.87 mm , 4 perimetry: 1.70 mm

**Vodorovné stěny**

- Plný vrstev: Vrchních:  Spodních:
- Minimální tloušťka skořepiny: Vrchních:  mm Spodních:  mm

Tloušťka vrchní skořepiny je 1.05 mm při výšce vrstvy 0.15 mm. Minimální tloušťka vrchní skořepiny je 0.7 mm.  
Tloušťka spodní skořepiny je 0.75 mm při výšce vrstvy 0.15 mm. Minimální tloušťka spodní skořepiny je 0.5 mm.

**Filament**

- Barva:
- Průměr:  mm
- Hustota:  g/cm<sup>3</sup>
- Náklady:  korun/kg
- Hmotnost cívky:  g

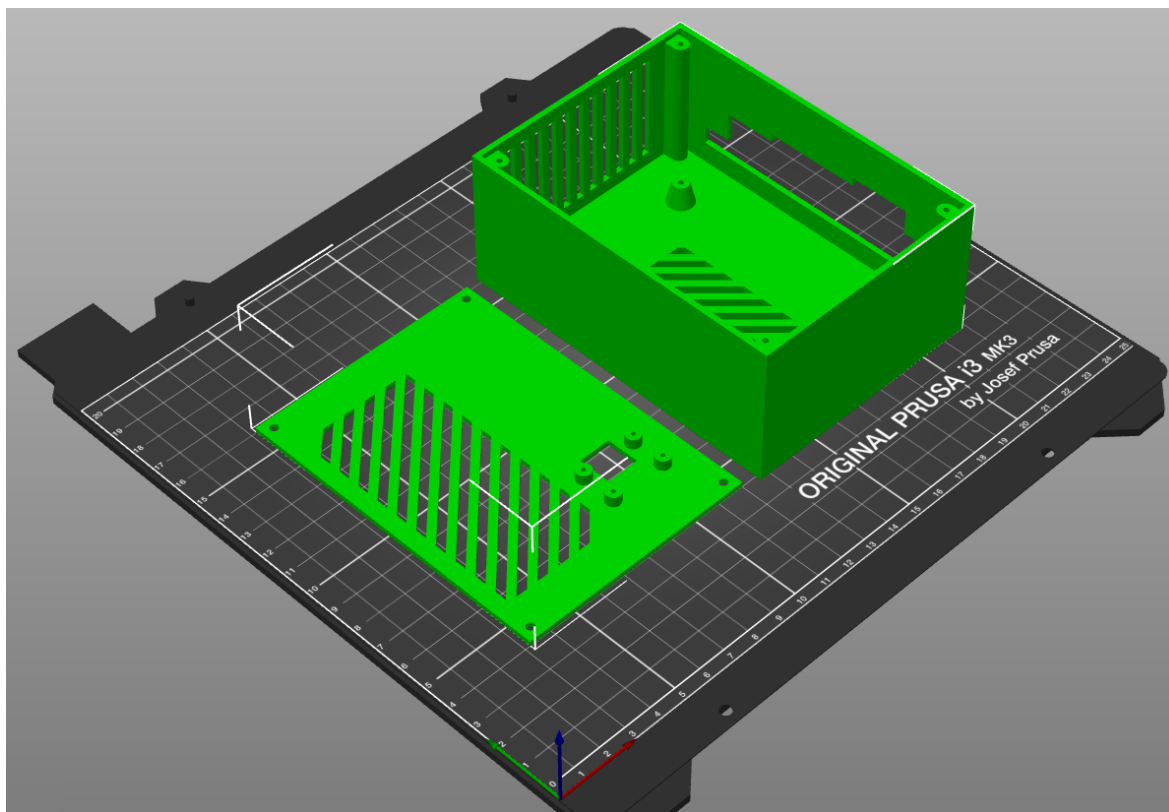
**Teplota**

- Tryska: První vrstva:  °C Ostatní vrstvy:  °C
- Tisková podložka: První vrstva:  °C Ostatní vrstvy:  °C

Obr. 25 – Nastavení 3D tisku krytu [autor]



Program PrusaSlicer také nabízí vizualizaci nakresleného modelu na tiskové ploše 3D tiskárny. Výslednou vizualizaci vymodelovaného krytu v programu lze vidět na Obr. 26.



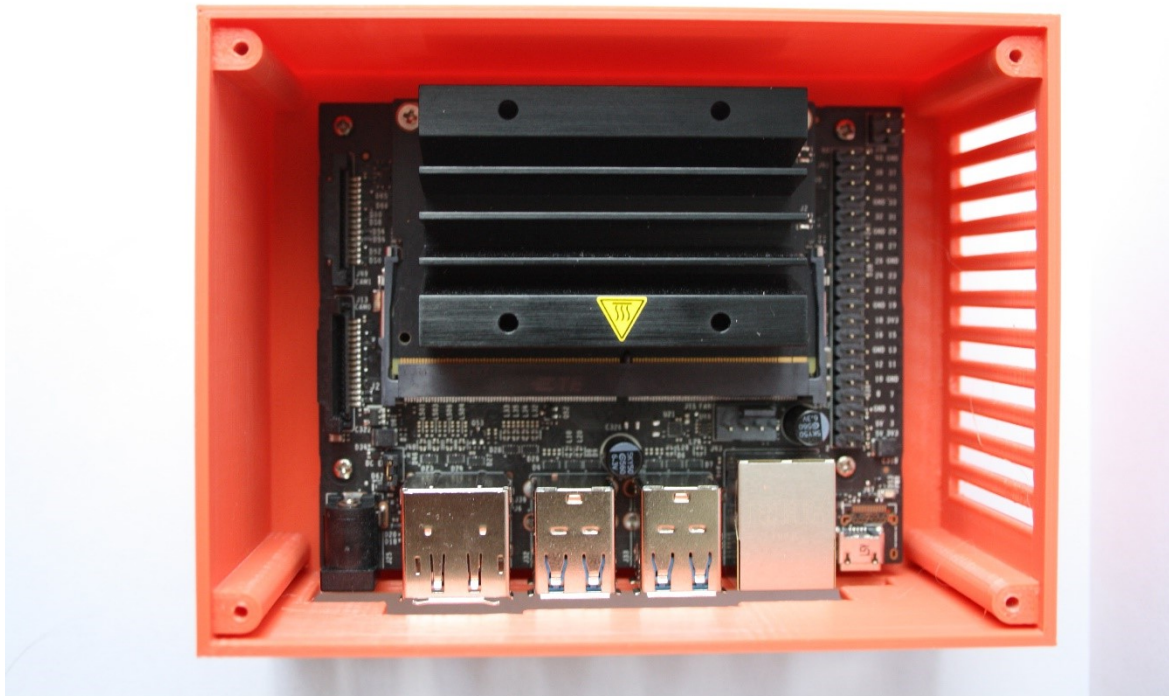
*Obr. 26 – Vizualizace vymodelovaného krytu [autor]*

Kompletní kryt se na 3D tiskárně tiskl přibližně 15 hodin a jeho rozměry činí 125×92×52 mm. Jak si lze všimnout z popisu tvorby krytu a přiložených snímků, jsou obě části krytu dostatečně pokryty větracími otvory. Důvodem je, že videoanalýza je pro počítač velmi náročným procesem a využívá většinu výkonu zařízení. Při využití výkonu takového rozsahu dochází ke značnému nárůstu teploty na zařízení. Velké množství větracích otvorů slouží pro přístup okolního vzduchu do zařízení a odvádění zahřátého vzduchu ze zařízení ven.

## **10.2 Montáž**

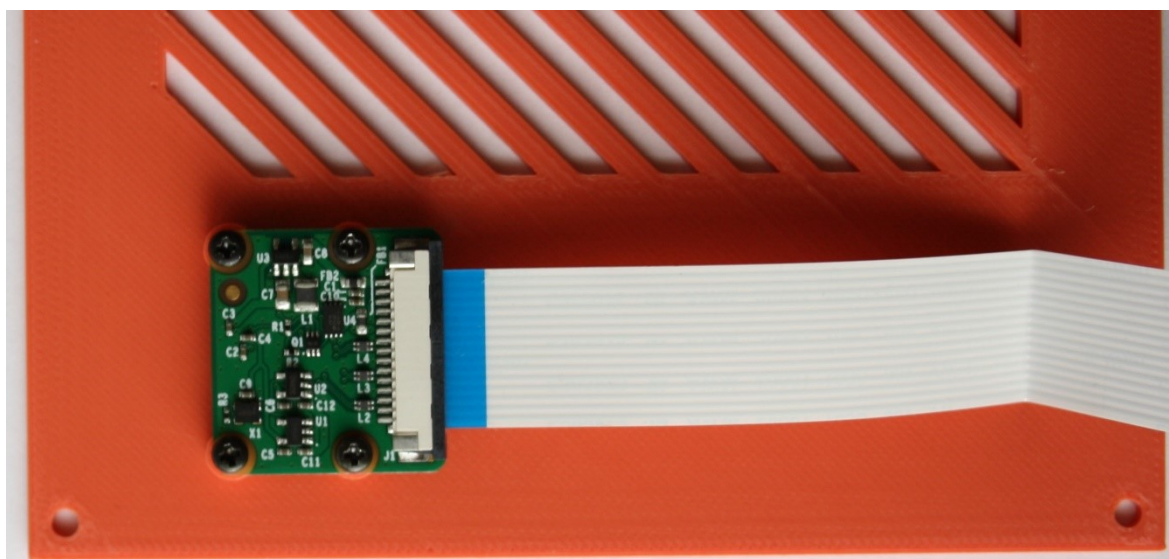
Dalším krokem v realizaci vlastního systému pro monitorování osob v objektu, byla montáž všech prvků do vytisknutého krytu a vytvoření tak jediného technického zařízení.

První částí montáže bylo uchycení počítače do hlavní části krytu. Deska počítače Jetson Nano je vybavena čtyřmi montážními otvory, kterými se za pomoci samořezných šroubů uchytila k distančním sloupkům na spodní straně krytu viz Obr. 27.



*Obr. 27 – Uchycení počítače k hlavní části krytu [autor]*

Dalším krokem montáže bylo uchycení kamery k vrchní části krytu. Stejně jako počítač i kamera je vybavena čtyřmi montážními otvory. Za pomoci samořezných šroubů byla kamera skrze montážní otvory uchycena k distančním sloupkům, které se vyskytují na spodní straně krytu viz Obr. 28.



*Obr. 28 – Uchycení kamery k vrchní části krytu [autor]*

Po uchycení kamery následovalo její připojení pomocí CSI konektoru k počítači. Jakmile byla kamera připojena, nic nebránilo poslednímu kroku montáže, a to spojení dvou částí krytu dohromady. Stejně jako u předchozích kroků bylo využito samořezných šroubů. Ty se skrze montážní otvory vrchní části zašroubovaly do distančních sloupků hlavní části krytu. Výsledek kompletní montáže a finální vzhled zařízení lze vidět na Obr. 29.



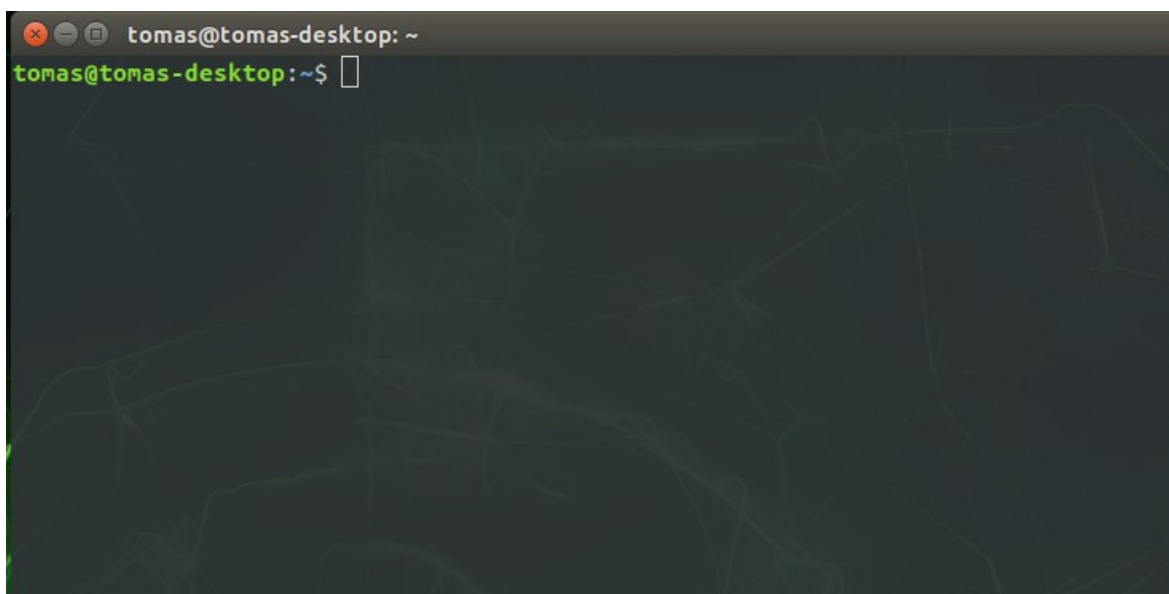
*Obr. 29 – Vzhled zařízení [autor]*

### 10.3 Instalace softwaru a spuštění zařízení

Dalším krokem realizace bylo nainstalování potřebného softwaru. Nejprve bylo nutné nainstalovat OS Ubuntu. Jelikož je Ubuntu open source platforma, lze jej zdarma stáhnout pomocí internetového prohlížeče. Stažený OS byl následně nainstalován na paměťovou kartu, která byla vložena do počítače Jetson Nano.

Veškerý další potřebný software se do počítače instaluje skrze Ubuntu, a to pomocí vestavěného programu nazývaného se Terminal. Program Terminal je obdoba příkazového řádku v OS Windows. Pro instalaci dalšího softwaru bylo tedy nutné spustit zařízení. Nejprve se k počítači pomocí vhodných konektorů připojila ovládací zařízení (klávesnice a myš), monitor a ethernetový kabel pro připojení k internetu. Jakmile byly všechny periférie připojeny, bylo do počítače přivedeno napájení a došlo k jeho spuštění. Při prvotním spuštění bylo nastaveno přístupové heslo do systému. Pomocí hesla zadávaného při každém spuštění je zařízení chráněné před neoprávněným použitím.

V již zapnutém počítači byl otevřen program Terminal a v něm pomocí textových příkazů došlo ke stažení a nainstalování vývojového prostředí IDLE Python. Vzhled programu Terminal lze vidět na Obr. 30. Dále se pomocí textových příkazů stáhl model neuronové sítě (SSD-MobileNet-v2) a programové knihovny jetson-inference a jetson-utils. Tímto byl nainstalován všechen potřebný software.

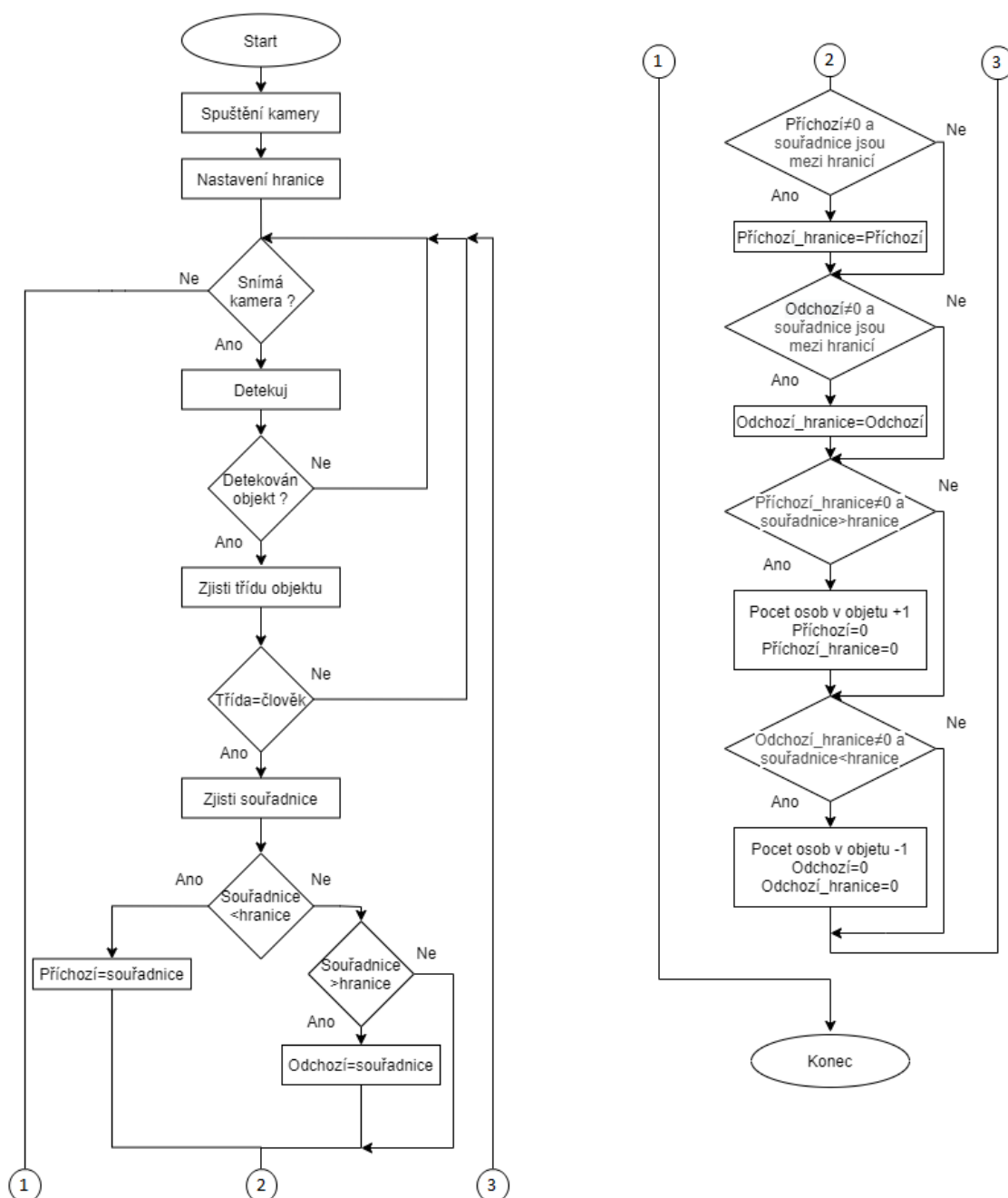


Obr. 30 – Program Terminal [autor]



## 10.4 Programování

Před zahájením programování videoanalytické aplikace byl vyhotoven vývojový diagram. Ten poskytuje stručný přehled o tom, jak bude celá aplikace fungovat a slouží jako vodítko při psaní programu. Vývojový diagram lze vidět na Obr. 31. Kompletní zdrojový kód vytvořeného programu je k vidění v Příloze č. 1.



Obr. 31 – Vývojový diagram programu [autor]

### 10.4.1 Implementace knihoven a vytvoření proměnných

Na úvod bylo nutné připojit k programu všechny potřebné knihovny. Kromě již dříve zmíněných knihoven `jetson-inference` a `jetson-utils` se jedná také o knihovnu `datetime`, která obsahuje různorodé funkce pro práci s časem. Připojení knihoven bylo provedeno pomocí příkazu `import`, jak lze vidět na Obr. 32.

```
1 |
2 | import jetson.inference
3 | import jetson.utils
4 | import datetime
5 |
```

*Obr. 32 – Připojení programových knihoven [autor]*

Po připojení knihoven následovalo vytvoření proměnných. Ty slouží k uložení hodnot nezbytných pro správnou funkci programu. V rámci realizovaného programu se jedná například o proměnnou hranice, třída objektu, souřadnice objektu, počet osob, čas atd. Na Obr. 33 jsou k vidění proměnné s názvem třída objektu. Tyto proměnné slouží k uložení číselné hodnoty, která identifikuje konkrétní třídu klasifikovaného objektu v obraze. Jak lze vidět, proměnná je zde uvedena pro šest objektů. Je to z toho důvodu, že program je vytvořen tak, aby byl schopen pracovat a počítat až s šesti objekty v jedné scéně současně. Ostatní proměnné vztahující se přímo k detekovaným objektům, jako jsou například souřadnice, musely být také vytvořeny šestkrát.

Veškeré proměnné jsou pevně stanovené a uživatel je není schopen upravit během spuštěného programu. Jedinou výjimkou je proměnná nazývaná se hranice. Ta je zadávána uživatelem z klávesnice při spuštění.

```
6 | Trida_objekt1=0
7 | Trida_objekt2=0
8 | Trida_objekt3=0
9 | Trida_objekt4=0
10 | Trida_objekt5=0
11 | Trida_objekt6=0
```

*Obr. 33 – Proměnné [autor]*

### 10.4.2 Nastavení rozlišení a programování hranice

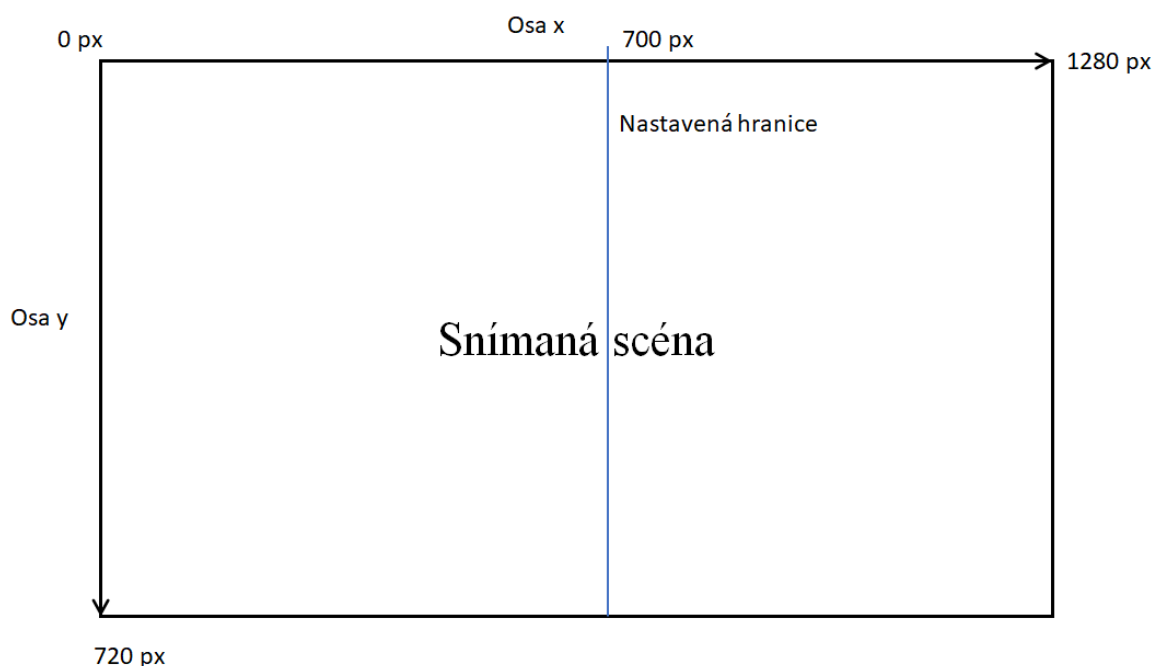
Rozlišení scény bylo nastaveno na 1280×720 px. Jedná se o výchozí rozlišení nastavené v knihovně `jetson-utils`. Kamera je sice schopna snímat ve větším rozlišení, a to 1920×1080 px, ovšem provádět detekci objektů a jejich klasifikaci je pro zařízení při takovém rozlišení značně obtížné. Výsledkem je rapidní úbytek FPS ve videosekvenci.

Kvalita snímaného obrazu ve výchozím rozlišení je pro potřeby vlastního systému dostatečná. Na Obr. 34 je vidět část kódu programu, která se zabývá nastavením hranice.

```
69| print("Velikost snímané scény je 1280x720px")
70| hranice = float(input("\nZadejte x souřadnici v pixelech;\n
71| pro stanovení hranice překročení: "))
72| print("\nByla zadána hranice :",hranice," px")
73| cas=datetime.datetime.now()
74| print("\nSpusteno v : ",cas)
```

*Obr. 34 – Nastavení hranice [autor]*

Program po spuštění nejprve upozorní uživatele na rozlišení scény v pixelech a následně jej vyzve k zadání hranice. Ta se zadává také v pixelech a jedná se vlastně o souřadnici osy x snímaného obrazu. Hodnota hranice je velmi důležitá. Při jejím překročení detekovanou osobou dojde k připočtení či odečtení celkového počtu osob v monitorovaném objektu. Správné nastavení hranice vzhledem ke konkrétní scéně je tedy kritické. Vizualizace scény ve výchozím rozlišení a nastavenou hranicí na 700 px lze vidět na Obr. 35.



*Obr. 35 – Vizualizace scény s nastavenou hranicí [autor]*

Po zadání hranice vypíše program zadanou hodnotu na obrazovku pro kontrolu. Na dalším řádku pak vypíše čas spuštění aplikace. Příkaz pro zjištění aktuálního času lze vidět na Obr. 34 řádek 73.

### 10.4.3 Implementace neuronové sítě, spuštění kamery a detekce

Dalším krokem v programování videoanalytické aplikace bylo implementování neuronové sítě, která bude provádět detekci a klasifikaci objektů. Implementace byla provedena pomocí funkce definované v knihovně jetson-inference. Na Obr. 36. lze vidět část programu, konkrétně řádek 80 až 86. Funkce pro připojení neuronové sítě je zobrazena na řádku 80.

```
80 net = jetson.inference.detectNet("ssd-mobilenet-v2", threshold=0.5)
81 camera = jetson.utils.videoSource("csi://0")
82 display = jetson.utils.videoOutput("display://0")
83
84 while display.IsStreaming():
85     img = camera.Capture()
86     detections = net.Detect(img)
```

*Obr. 36 – Implementace neuronové sítě a spuštění kamery [autor]*

Řádek 81 a 82 má na starost připojení a spuštění kamery. Na řádku 84 je potom vytvořen cyklus, jehož podmínkou je spuštěná snímající kamera. Jedná se o velmi důležitý řádek programu, jelikož veškerá videoanalýza bude probíhat uvnitř tohoto cyklu. Podmínka cyklu nám říká, že dokud bude kamera spuštěna a bude snímat, bude probíhat také videoanalýza. Ve chvíli, kdy bude snímání obrazu ukončeno, dojde k nesplnění podmínky cyklu a videoanalýza se ukončí. Během jednoho průběhu programu zmíněným cyklem pracuje program s jediným snímkem zachyceným kamerou.

Řádek 85 je prvním řádkem cyklu a jeho úkolem je uložit do proměnné jednotlivé snímky z videosekvence. Na následujícím řádku je proměnná s uloženým snímkem použita ve funkci Detect. Jedná se o funkci, která za pomoci neuronové sítě provede v jednotlivém snímku detekci objektů a jejich klasifikaci.

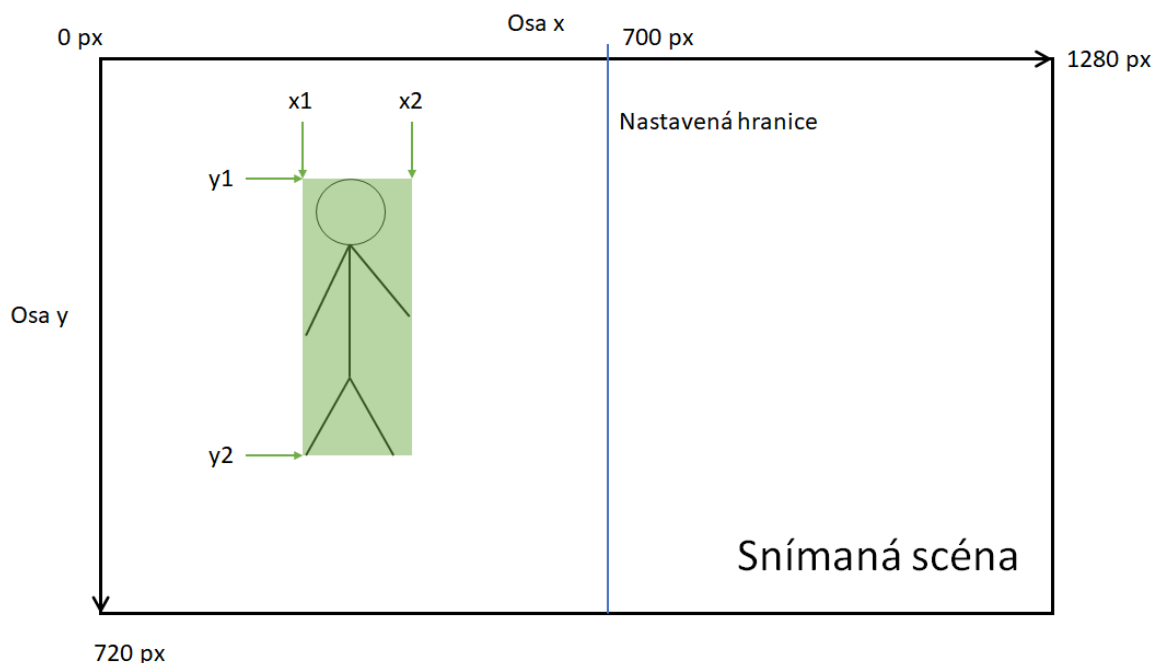
### 10.4.4 Zjištění třídy objektu a souřadnic

S funkčním snímáním, detekcí a klasifikací bylo na řadě zajištění potřebných informací o detekovaných objektech. K získání těchto informací byla využita funkce Detection. Ta slouží k výpisu veškerých vlastností, které jsou přiřazené detekovaným objektům. K realizaci systému bylo potřeba zajistit informace o třídě objektu a jeho souřadnicích.

Zjištění souřadnic detekovaného objektu je stěžejní. Pomocí nich je totiž sledována poloha objektu ve scéně, a tím i moment překročení hranice. Každý detekovaný objekt



disponuje čtyřmi souřadnicemi, levou, pravou, horní a dolní. Souřadnice jsou objektu přiděleny pomocí obdélníku, který je vytvořen při detekci a tvoří pomyslný rám okolo objektu viz Obr. 37.



Obr. 37 - Souřadnice detekovaného objektu [autor]

Pro získání informací o třídě a souřadnicích všech objektů, vyskytujících se na snímku byly použito dalších cyklů. Na Obr. 38 je k vidění část cyklu, který získává levé souřadnice (označené jako x1) detekovaných objektů.

```

120     for detection in detections:
121         l=detection.Left
122         if x1_objekt1 == 0:
123             x1_objekt1=l
124             continue
125         if x1_objekt1 != 0 and x1_objekt2 == 0:
126             x1_objekt2=l
127             continue
128         if x1_objekt1 != 0 and x1_objekt2 != 0 and x1_objekt3 == 0:
129             x1_objekt3=l

```

Obr. 38 – Cyklus pro získání souřadnic [autor]

### 10.4.5 Počítání osob

V této fázi byl program již aplikací, která byla schopna využívat dvou videoanalytických funkcí, a to detekce a klasifikace objektů za pomoci neuronové sítě.

Dalším krokem byla realizace videoanalytické funkce počítání osob. Počítání osob ke své funkci využívá detekci a klasifikaci objektů. Cílem funkce je udržovat neustálý

přehled o počtu osob, vyskytujících se v objektu nebo zabezpečené zóně. Princip této videoanalytické funkce je založen na překročení nastavené hranice ve scéně. Hranice vlastně rozděluje snímanou scénu na dvě strany. Jedna strana, na které se vyskytuje vstup do objektu je stranou příchozí, ta druhá je pak stranou odchozí. Pokud osoba vstupuje do objektu, je detekována na straně příchozí a následným překročením hranice je připočtena k celkovému počtu osob uvnitř. Naopak osoba odcházející je nejprve detekována na straně odchozí a následným překročením hranice je odečtena od celkového počtu osob v objektu. Počítání osob je v programu řešeno několika podmíněnými příkazy viz řádek 165 až 192 na Obr. 39.

```

165     if x1_objekt1 != 0 and Trida_objekt1 == 1:
166         if x1_objekt1 < hranice and x2_objekt1 < hranice:
167             px1_objekt1=x1_objekt1
168             px2_objekt1=x2_objekt1
169         elif x1_objekt1 > hranice and x2_objekt1 > hranice:
170             ox1_objekt1=x1_objekt1
171             ox2_objekt1=x2_objekt1
172
173         if px1_objekt1 !=0:
174             if x1_objekt1 < hranice and x2_objekt1 > hranice:
175                 prichod1=1
176                 if x1_objekt1 > hranice and prichod1 == 1:
177                     pocitac +=1
178                     px1_objekt1=0
179                     px2_objekt1=0
180                     ox1_objekt1=0
181                     ox2_objekt1=0
182                     prichod1=0
183         if ox2_objekt1 !=0:
184             if x1_objekt1 < hranice and x2_objekt1 > hranice:
185                 odchod1=1
186                 if x2_objekt1 < hranice and odchod1 == 1:
187                     pocitac -=1
188                     ox1_objekt1=0
189                     ox2_objekt1=0
190                     px1_objekt1=0
191                     px2_objekt1=0
192                     odchod1=0

```

*Obr. 39 – Programování funkce počítání osob [autor]*

První podmínka zjišťuje, zda je detekován objekt a je-li klasifikován jako člověk. Pokud je podmínka splněna, přikročí se k další, která má za úkol identifikaci polohy osoby vzhledem k stanovené hranici. V tuto chvíli je již známo, kde se osoba ve scéně nachází a je jí přidělen status příchozí nebo odchozí. Následuje podmíněný příkaz, který má za úkol kontrolovat výskyt příchozí či odchozí osoby na hranici. Tento stav nastane, když má osoba levou souřadnici na jedné straně hranice a tu pravou souřadnici na druhé. Poslední podmíněný příkaz ověřuje, zdali příchozí či odchozí osoba byla na hranici a zdali je její

současná poloha za ní. Splněním poslední podmínky dojde k připočtení či odečtení z celkového počtu osob v objektu a dojde k vynulování použitých proměnných.

### 10.4.6 Datový výstup

Výstup dat z videoanalytické aplikace byl realizován třemi způsoby. Prvním a druhým způsobem je výstup na monitor ve formě grafických a textových dat. Třetím způsobem je také textový formát, ale tentokrát uložen do souboru.

Grafický výstup na monitor byl proveden následujícími příkazy viz Obr. 40. První příkaz vytvoří na monitoru okno, ve kterém je zobrazena snímaná scéna kamerou v reálném čase. Druhý příkaz slouží k popisku vyvolaného okna a zobrazuje aktuální FPS videosekvence.

```
395|         display.Render(img)
396|
415|
416|         display.SetStatus("Monitoring osob v objektu ;\
417|         | Network {:.0f} FPS".format(net.GetNetworkFPS()))
```

*Obr. 40 – Programování grafického výstupu [autor]*

Textový výstup na monitor byl také realizován pomocí okna, zobrazujícího se vedle grafického výstupu. V textovém okně, je při spuštění zobrazena výzva pro zadání hranice a je zde uveden čas spuštění aplikace. Dále se v textovém okně neustále zobrazují aktuální informace o počtu osob v objektu. Tyto informace jsou doplněny o časový údaj, kdy došlo ke změně počtu osob. Textový výstup na monitor je proveden pomocí následujících příkazů viz Obr. 41. Prvním příkazem je podmínka, která reaguje na změnu počtu osob. Pokud tedy dojde ke zvýšení či snížení počtu osob v objektu, podmínka je splněna a program tuto informaci vypíše na monitor a doplní ji o časový údaj kdy k tomu došlo.

```
88|         if pocet_osob != pocitac:
89|             pocet_osob = pocitac
90|             cas=datetime.datetime.now()
91|             print(cas, " Pocet osob v objektu: ",pocet_osob)
```

*Obr. 41 – Programování textového výstupu na monitor [autor]*

Textový výstup do souboru byl realizován obdobně jako textový výstup na monitor. V programu je řešen stejnou podmínkou s tím rozdílem, že program nic na monitor nevypisuje, ale vytvoří soubor textového formátu. Tomuto souboru je udělen název Pocet\_osob a je do něj uložen čas spuštění aplikace a veškeré informace o počtech osob

v objektu i s časovými údaji. Soubor tedy funguje jako historie aplikace a lze jej kdykoliv otevřít a zkontrolovat v něm uložená data. Textový formát souboru je kompatibilní se všemi známými softwary a lze jej otevřít na jakékoliv platformě. Část programu, která slouží k vytvoření a zápisu informací do souboru je k vidění na Obr. 42.

```
92 |         soubor = open("Pocet_osob.txt", "a")
93 |         soubor.write("\n")
94 |         soubor.write(str(cas))
95 |         soubor.write(" Pocet osob v objektu: ")
96 |         soubor.write(str(pocet_osob))
97 |         soubor.close()
```

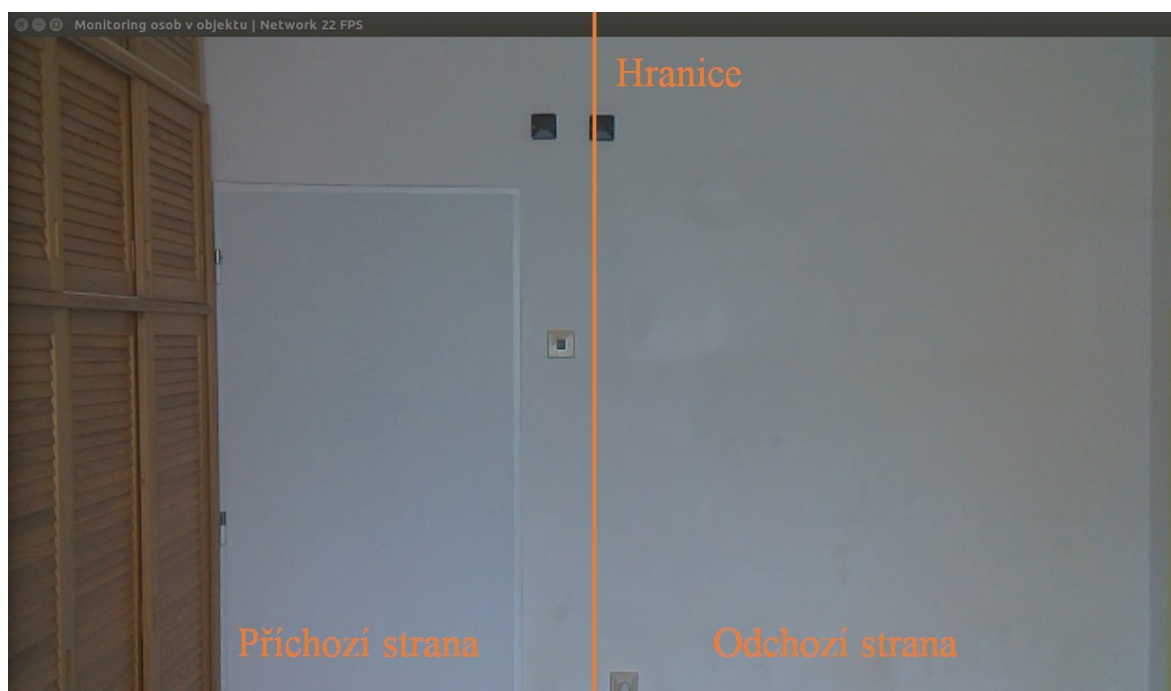
*Obr. 42 – Programování textového výstupu do souboru [autor]*

## 11 Testování

Po realizaci systému pro monitorování osob v objektu bylo nutné otestovat jeho funkčnost. Kromě funkčnosti samotné videoanalytické aplikace byla testována i tepelná námaha hardwarové části zařízení.

### 11.1 Testování videoanalytické aplikace

Nejprve bylo zařízení vhodně situováno, aby kamera ideálně zabírala scénu. Následně byla aplikace spuštěna a došlo k nastavení hranice ve scéně. Ta byla nastavena na hodnotu 640 px, tedy přesně doprostřed snímaného obrazu. Jako příchozí strana je v tomto případě v programu nastavena strana vlevo od hranice. Snímanou scénu spolu s naznačením hranice a popisem stran lze vidět na Obr. 43.



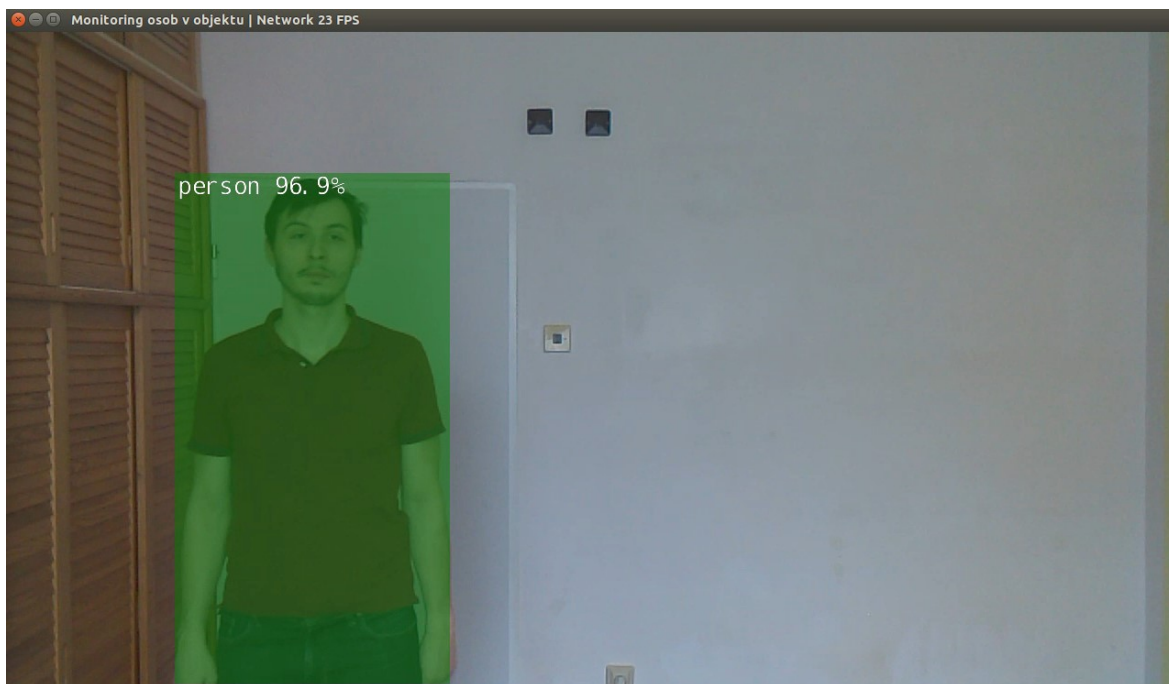
Obr. 43 – Snímaná scéna [autor]

V tuto chvíli vypadal textový výstup na monitoru následovně viz Obr. 44.

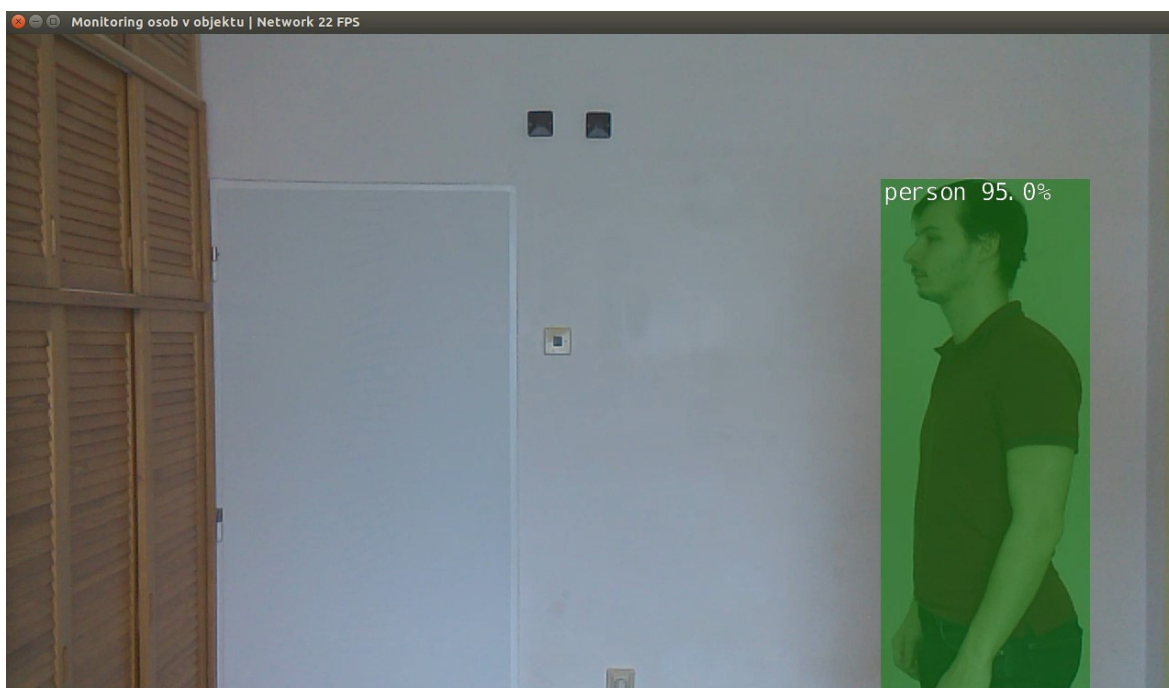
```
Velikost snímané scény je 1280x720px
Zadejte x souřadnici v pixelech pro stanovení hranice překročení:
640
Byla zadána hranice : 640.0 px
Spusteno v : 2021-03-12 20:02:33.861361
```

Obr. 44 – Textový výstup po spuštění a nastavení

Dalším krokem bylo otestování detekce a klasifikace objektů. Výsledek je zobrazen na následujících snímcích viz Obr. 45, Obr. 46. Na zmíněných snímcích lze vidět detekovaný



*Obr. 45 – Detekce a klasifikace objektů 1 [autor]*



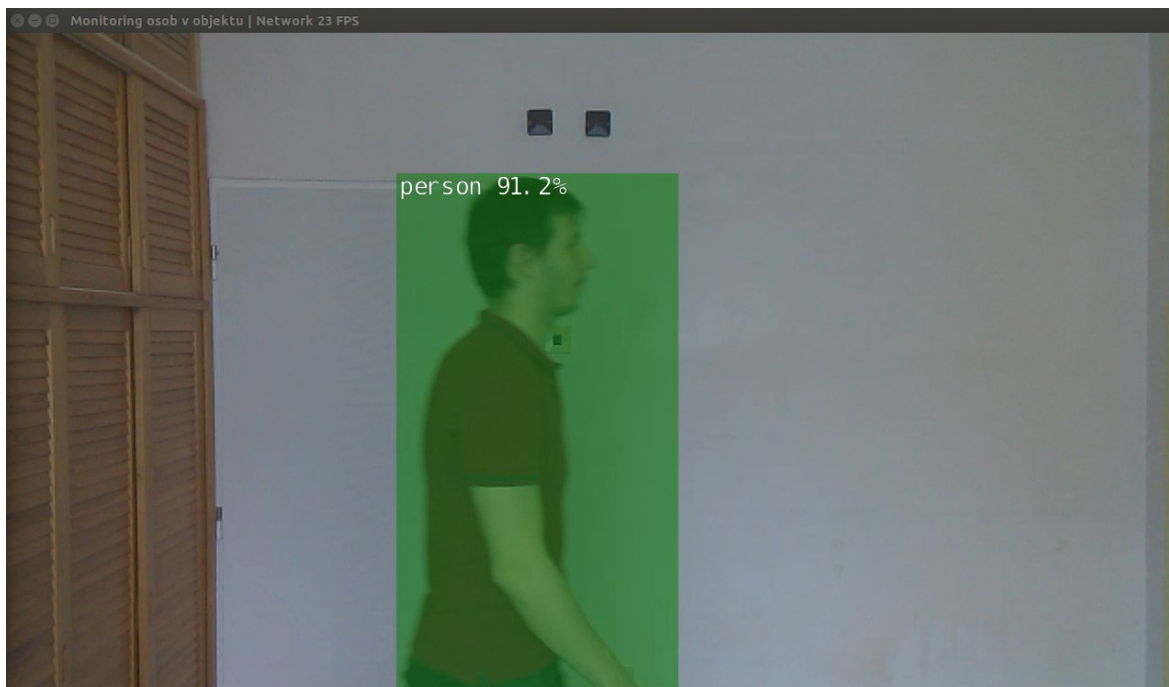
*Obr. 46 – Detekce a klasifikace objektů 2 [autor]*

objekt z přední strany a z profilu. Kolem obvodu detekovaného objektu je vykreslený rámeček určující souřadnice objektu. V levém horním rohu rámečku je zobrazena výsledná klasifikace objektu spolu s procentuální hodnotou. Tato hodnota určuje na kolik procent



si je systém jist, že klasifikoval daný objekt správně. Během testování byla tato procentuální hodnota za dobrých světelných podmínek (během dne) vždy nad hodnotou 90 %. Při zhoršení světelných podmínek (šero, umělé osvětlení) klesla hodnota mezi 70 až 80 %.

Dále následovalo testování funkce počítání osob. Detekovaná osoba, která se vyskytovala na příchozí straně, byla po překročení nastavené hranice přičtena k celkovému počtu osob v objektu. Naopak detekovaná osoba vyskytující se na straně odchozí byla po překročení hranice odečtena od celkového počtu osob. Při okamžiku překročení hranice osobou byla tato informace okamžitě vypsaná v textové podobě na monitor a uložena do souboru spolu s časovým údajem. Moment překročení stanovené hranice detekovanou osobou lze vidět na Obr. 47.



*Obr. 47 – Překročení hranice [autor]*

Pro otestování funkce počítání osob bylo provedeno celkem 30 testů. Během jednoho testu byla aplikace vždy spuštěna minimálně po dobu 8 hodin. Při testování byl výsledný počet osob v objektu vždy spočten správně. Kamera během testování snímala průměrnou hodnotou 23 FPS.

Ukázku textového výpisu na monitor po detekci a započítání několika osob lze vidět na Obr. 48.

```
Velikost snímané scény je 1280x720px

Zadejte x souřadnici v pixelech pro stanovení hranice překročení:
640

Byla zadána hranice : 640.0 px

Spusteno v : 2021-03-12 20:02:33.861361
2021-03-12 20:04:02.490398 Pocet osob v objektu: 1
2021-03-12 20:04:14.301870 Pocet osob v objektu: 0
2021-03-12 20:04:23.169620 Pocet osob v objektu: 1
2021-03-12 20:04:35.105270 Pocet osob v objektu: 0
2021-03-12 20:04:54.179034 Pocet osob v objektu: 1
2021-03-12 20:05:41.499604 Pocet osob v objektu: 0
2021-03-12 20:06:27.976450 Pocet osob v objektu: 1
2021-03-12 20:06:39.655793 Pocet osob v objektu: 0
2021-03-12 20:06:54.961150 Pocet osob v objektu: 1
2021-03-12 20:07:17.073311 Pocet osob v objektu: 2
2021-03-12 20:07:28.851910 Pocet osob v objektu: 3
2021-03-12 20:07:45.776644 Pocet osob v objektu: 4
2021-03-12 20:08:04.986839 Pocet osob v objektu: 5
2021-03-12 20:08:49.425278 Pocet osob v objektu: 4
2021-03-12 20:09:01.242819 Pocet osob v objektu: 5
2021-03-12 20:09:30.710300 Pocet osob v objektu: 4
2021-03-12 20:09:43.302943 Pocet osob v objektu: 5
2021-03-12 20:10:20.520998 Pocet osob v objektu: 4
2021-03-12 20:10:20.641800 Pocet osob v objektu: 3
2021-03-12 20:10:25.598688 Pocet osob v objektu: 4
2021-03-12 20:10:43.844997 Pocet osob v objektu: 3
2021-03-12 20:10:53.271006 Pocet osob v objektu: 2
2021-03-12 20:11:11.596947 Pocet osob v objektu: 3
2021-03-12 20:11:29.534774 Pocet osob v objektu: 2
2021-03-12 20:12:34.436236 Pocet osob v objektu: 3
2021-03-12 20:13:05.178341 Pocet osob v objektu: 2
2021-03-12 20:13:14.256038 Pocet osob v objektu: 3
```

Ln: 13 Col: 0

Obr. 48 – Textový výpis na monitor [autor]

## 11.2 Testování tepelného namáhání

Vzhledem k tomu, že videoanalýza je pro počítač velmi náročný proces, dochází během její funkce ke značnému nárůstu teploty. Především na zatíženém grafickém čipu a procesoru. Uzavřením počítačového hardwaru do krytu, může dojít ke strmějšímu nárůstu teploty. Z tohoto důvodu bylo provedeno testování tepelné zátěže na zařízení.



Počítač Jetson Nano je vybaven teplotními čidly, které snímají teplotu na grafickém čipu a procesoru. Dojde-li k překročení hraniční teploty, počítač přejde do stavu nouzového vypnutí. Tato teplota je pro procesor nastavena na 102,5 C°, a pro grafický čip na 103 C°. [18]

Testování tepelného namáhání zařízení proběhlo v pokojové teplotě. Během testu byla teplota z čidel odečítána každou hodinu po dobu deseti hodin. Výsledky testování lze vidět v Tab. 4. Z tabulky lze vyčíst, že na zařízení nedochází k přehřívání. Přibližně po hodině od zapnutí dojde k ustálení teploty na grafickém čipu i procesoru. V následujících hodinách teplota osciluje kolem této ustálené hodnoty. Maximální zaznamenaná hodnota teploty na grafickém čipu činila 78 C°, na procesoru pak 80,5 C°.

*Tab. 4 – Testování tepelného namáhání zařízení [autor]*

Čas	Teplota Grafický čip	Teplota Procesor
Po zapnutí	30,5 C°	32,5 C°
1 hodina	77,5 C°	80,0 C°
2 hodina	77,0 C°	78,5 C°
3 hodina	73,0 C°	75,5 C°
4 hodina	70,5 C°	72,5 C°
5 hodina	75,0 C°	78,0 C°
6 hodina	78,0 C°	80,5 C°
7 hodina	72,0 C°	74,5 C°
8 hodina	75,0 C°	77,5 C°
9 hodina	77,5 C°	80,0 C°
10 hodina	76,0 C°	77,0 C°

## Závěr

Cílem diplomové práce bylo navrhnout a realizovat systém pro monitorování přítomnosti osob v objektu, založený na principu videoanalytických funkcí. K realizaci mělo být použito produktů open source platformy.

Diplomová práce je rozdělena na dvě části, teoretickou a praktickou. Teoretická část na svém počátku uvádí definici základních pojmů, následně se zabývá základy zpracování obrazu a tématu digitalizace. Dále teoretická část popisuje segmentaci obrazu a různé segmentační techniky, klasifikaci obrazu se zaměřením na neuronové sítě a definuje pojem videoanalýza. Závěrem teoretické části jsou uvedeny jednotlivé videoanalytické funkce se zaměřením na bezpečnostní aplikace a jejich použití v praxi.

Praktická část diplomové práce je zaměřena na návrh a tvorbu vlastního systému pro monitorování osob v objektu. V textu praktické části jsou nejprve identifikovány požadavky na vlastní systém spolu s principem jeho funkce. Následuje podrobný popis výběru vhodného technického řešení, jeho montáže a realizace softwarové části. K realizaci byl použit jednodeskový počítač Jetson Nano, který je mozkiem celého systému. Ke snímání obrazu byla použita kamera Raspberry pi V2. Kyt celého zařízení byl navrhnout, vymodelován a následně vytvořen na 3D tiskárně. Software byl realizován za pomoci produktů open source platformy. Princip funkce vlastního systému se opírá o videoanalytické funkce, kterými jsou detekce objektů, klasifikace objektů a počítání osob. Uvedené videoanalytické funkce byly realizovány za pomoci umělé neuronové sítě. Kompletní program vytvořené softwarové aplikace obsahuje 416 řádků. Součástí aplikace je také datový výstup, a to ve formě grafického a textového zobrazení dat na monitoru. V poslední řadě disponuje aplikace také uložením dat do historie ve formátu textového souboru.

Realizovaný systém je vhodný pro použití u vstupu do objektu. Díky videoanalytické funkci počítání osob je systém schopen mít neustálý přehled o počtu osob vyskytujících se uvnitř objektu. Tato informace nachází využití například při evakuaci osob z objektu nebo pro omezení vstupu určitého počtu osob do objektu. Výhodou realizovaného systému je malá finanční náročnost, a to díky využití open source produktů. Finanční náklady na pořízení nepřesahují částku 3000 Kč. Další výhodou systému je možnost snadného rozšíření a vylepšení. Díky těmto výhodám má systém perspektivní využití.

## Seznam zdrojů

- [1] A Guide to Video Analytics: Applications and Opportunities. *Tryo labs* [online]. [cit. 2021-02-17]. Dostupné z: <https://tryolabs.com/resources/video-analytics-guide/>
- [2] *Automa: časopis pro automatizační techniku*. 17. Praha: FCC Public., 1994-, 64 s. ISSN 1210-9592. Dostupné také z: <https://automa.cz/page-flip/casopis/automa/2011/02/index.html#page/1>
- [3] Concept of Pixel: Pixel. *JavaTpoint* [online]. [cit. 2021-02-13]. Dostupné z: <https://www.javatpoint.com/concept-of-pixel>
- [4] Convolutional Neural Networks. *IBM* [online]. 20.10.2020 [cit. 2021-02-13]. Dostupné z: [ibm.com/cloud/learn/convolutional-neural-networks](https://ibm.com/cloud/learn/convolutional-neural-networks)
- [5] Creating a more secure world and new value for businesses: with Video analytics. *Bosch* [online]. Únor 2020 [cit. 2021-02-22]. Dostupné z: [https://media.boschsecurity.com/fs/media/pb/media/support\\_1/tools/analytics\\_example\\_movies.pdf](https://media.boschsecurity.com/fs/media/pb/media/support_1/tools/analytics_example_movies.pdf)
- [6] Deep Learning. *IBM* [online]. 1.5.2020 [cit. 2021-02-11]. Dostupné z: <https://www.ibm.com/cloud/learn/deep-learning>
- [7] GANGULY, Ranita. Top 8 interesting facts about the human eye. In: *HealthSite* [online]. 5.3.2016 [cit. 2021-02-21]. Dostupné z: <https://www.thehealthsite.com/diseases-conditions/top-8-interesting-facts-about-the-human-eye-179180/>
- [8] HLAVÁČ, Václav a Miloš SEDLÁČEK. *Zpracování signálů a obrazů*. Praha: Vydavatelství ČVUT, 2000. ISBN 80-01-02114-9.
- [9] HLAVÁČ, Václav. *Digitální zpracování obrazu × počítačové vidění* [online]. Praha [cit. 2021-02-18]. Dostupné z: <http://people.ciirc.cvut.cz/~hlavac/TeachPresCz/11DigZprObr/012IntroDigImProcVsCompVisionCz.pdf>
- [10] Intelligent AI Video Analytics. *IntelliVision: AI and Video Analytics for Smart Cameras* [online]. [cit. 2021-02-17]. Dostupné z: <https://www.intellivision.com/intelligent-video-analytics/>
- [11] *IT-slovník: operační systém* [online]. [cit. 2021-01-05]. Dostupné z: <https://it-slovník.cz/pojem/operacni-system>

- [12] *Jetson modules: Technical specification* [online]. [cit. 2021-03-16]. Dostupné z: <https://developer.nvidia.com/embedded/jetson-modules>
- [13] KOLEKAR, Maheshkumar. *Intelligent Video Surveillance Systems: An Algorithmic Approach*. Boca Raton, Florida: Taylor & Francis Group, 2018. ISBN 978-1-4987-6711-8.
- [14] KONONOW, Piotr. What is Metadata (with examples). *Dataedo* [online]. 16.9.2018 [cit. 2021-02-17]. Dostupné z: <https://dataedo.com/kb/data-glossary/what-is-metadata>
- [15] Machine Learning. *IBM* [online]. 15.7.2020 [cit. 2021-02-11]. Dostupné z: <https://www.ibm.com/cloud/learn/machine-learning>
- [16] *Mioweb: Co je to open source* [online]. [cit. 2021-01-05]. Dostupné z: <https://www.mioweb.cz/slovnicek/open-source/>
- [17] Neural Networks. *IBM* [online]. 17.8.2020 [cit. 2021-02-11]. Dostupné z: <https://www.ibm.com/cloud/learn/neural-networks>
- [18] *Nvidia Jetson Linux Developer Guide* [online]. [cit. 2021-03-26]. Dostupné z: [https://docs.nvidia.com/jetson/l4t/index.html#page/Tegra%20Linux%20Driver%20Package%20Development%20Guide/power\\_management\\_nano.html#wwpID0E05J0HA](https://docs.nvidia.com/jetson/l4t/index.html#page/Tegra%20Linux%20Driver%20Package%20Development%20Guide/power_management_nano.html#wwpID0E05J0HA)
- [19] OBJECT DETECTION IN COMPUTER VISION. *AlwaysAI* [online]. [cit. 2021-02-21]. Dostupné z: <https://learn.alwaysai.co/object-detection>
- [20] *OpenCV: Canny Edge Detection* [online]. [cit. 2021-02-05]. Dostupné z: [https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_imgproc/py\\_canny/py\\_canny.html](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_canny/py_canny.html)
- [21] *Python v ČR: python* [online]. [cit. 2021-01-05]. Dostupné z: <https://python.cz/>
- [22] *Python: What is Python? Executive Summary* [online]. [cit. 2021-03-20]. Dostupné z: <https://www.python.org/doc/essays/blurb/>
- [23] *RPishop.cz: Raspberry Pi kamera V2* [online]. [cit. 2021-03-16]. Dostupné z: <https://rpishop.cz/kategorie/329-raspberry-pi-kamera-modul-v2.html>
- [24] *Securia PRO: co je IP kamera* [online]. 2020 [cit. 2021-01-06]. Dostupné z: <https://www.securiapro.cz/clanek/co-je-ip-kamera/>

- [25] ŠEVČÍK, Jiří. Princip činnosti, typy a komunikační rozhraní IP kamer. *Tzbinfo* [online]. 2013 [cit. 2021-02-21]. Dostupné z: <https://elektro.tzb-info.cz/10480-princip-cinnosti-typy-a-komunikacni-rozhrani-ip-kamer>
- [26] ŠPANĚL, Michal a Vítěslav BERAN. *Obrazové segmentační techniky: Přehled existujících metod* [online]. 12.10.2005 [cit. 2021-02-02]. Dostupné z: <http://www.fit.vutbr.cz/~spanel/segmentace/>
- [27] *Tzbinfo: kamerové systémy* [online]. [cit. 2021-01-05]. Dostupné z: <https://www.tzb-info.cz/kamerove-systemy>
- [28] Video analytics. *AXIS communication* [online]. [cit. 2021-02-14]. Dostupné z: <https://www.axis.com/learning/web-articles/video-analytics/what-are-video-analytics>
- [29] Video Analytics. *Senstar* [online]. [cit. 2021-02-17]. Dostupné z: <https://senstar.com/products/video-analytics/>

## Seznam obrázků

Obr. 1 – Obrázek oka, s viditelnou maticí pixelů [autor, upraveno z [7]] .....	6
Obr. 2 – a) skoková, b) liniová, c) šikmá, d) střešní [13] .....	10
Obr. 3 – Původní obraz a obraz po segmentaci [20].....	11
Obr. 4 – a) původní obraz, b) detekce hran, c) spojování hran [2] .....	12
Obr. 5 – Histogram obrazu a jeho prahování [autor, upraveno z [13]] .....	13
Obr. 6 – Princip metody dělení a spojování oblastí [autor, upraveno z [13]].....	14
Obr. 7 – Ukázka klasifikace objektů [19] .....	16
Obr. 8 – Struktura umělé neuronové sítě [autor, převzato z [17]] .....	18
Obr. 9 – Použití filtru na vstupní obraz v konvoluční vrstvě sítě [4] .....	20
Obr. 10 – Videoanalytická funkce překročení hranice [5] .....	24
Obr. 11 – Detekce narušení oblasti [5] .....	24
Obr. 12 – Videoanalytická funkce počítání lidí [5] .....	25
Obr. 13 – Detekce opuštěných předmětů [5] .....	26
Obr. 14 – Jednodeskový počítač NVIDIA Jetson Nano [autor] .....	29
Obr. 15 – Paměťová karta připravená k zasunutí do slotu [autor].....	30
Obr. 16 – Kamera Raspberry Pi V2 [autor] .....	31
Obr. 17 – Umístění sousedního napájecího konektoru [autor] .....	32
Obr. 18 – Napájecí adaptér [autor] .....	33
Obr. 19 – Paměťová karta MicroSDHC [autor] .....	34
Obr. 20 – Hlavní část krytu [autor].....	35
Obr. 21 – Vrchní část krytu [autor] .....	35
Obr. 22 – Grafické rozhraní operačního systému Ubuntu [autor] .....	36
Obr. 23 – Vývojové prostředí IDLE Python [autor].....	37
Obr. 24 – Model krytu nakreslený v programu AutoCAD [autor] .....	40
Obr. 25 – Nastavení 3D tisku krytu [autor] .....	41
Obr. 26 – Vizualizace vymodelovaného krytu [autor] .....	42
Obr. 27 – Uchycení počítače k hlavní části krytu [autor].....	43
Obr. 28 – Uchycení kamery k vrchní části krytu [autor] .....	43
Obr. 29 – Vzhled zařízení [autor] .....	44
Obr. 30 – Program Terminal [autor].....	45
Obr. 31 – Vývojový diagram programu [autor].....	46

Obr. 32 – Připojení programových knihoven [autor] .....	47
Obr. 33 – Proměnné [autor] .....	47
Obr. 34 – Nastavení hranice [autor] .....	48
Obr. 35 – Vizualizace scény s nastavenou hranicí [autor].....	48
Obr. 36 – Implementace neuronové sítě a spuštění kamery [autor] .....	49
Obr. 37 - Souřadnice detekovaného objektu [autor].....	50
Obr. 38 – Cyklus pro získání souřadnic [autor].....	50
Obr. 39 – Programování funkce počítání osob [autor] .....	51
Obr. 40 – Programování grafického výstupu [autor].....	52
Obr. 41 – Programování textového výstupu na monitor [autor].....	52
Obr. 42 – Programování textového výstupu do souboru [autor] .....	53
Obr. 43 – Snímaná scéna [autor] .....	54
Obr. 44 – Textový výstup po spuštění a nastavení .....	54
Obr. 45 – Detekce a klasifikace objektů 1 [autor] .....	55
Obr. 46 – Detekce a klasifikace objektů 2 [autor] .....	55
Obr. 47 – Překročení hranice [autor] .....	56
Obr. 48 – Textový výpis na monitor [autor].....	57

## Seznam tabulek

Tab. 1 – Popisy jednotlivých částí počítače NVIDIA Jetson Nano [autor].....	29
Tab. 2 – Technické parametry počítače Jetson Nano [12].....	30
Tab. 3 – Technické parametry kamery Raspberry Pi V2 [23].....	32
Tab. 4 – Testování tepelného namáhání zařízení [autor].....	58



## **Seznam příloh**

Příloha č. 1 – Zdrojový kód programu